Université de Strasbourg

ICUBE
Laboratoire des sciences de l'ingénieur de l'informatique et de l'imagerie

École Doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur

## Habilitation à Diriger des Recherches

présentée par

Julien Montavont

# Mobility Support in Low Power Asynchronous Wireless Networks

Soutenance prévue le 29 novembre 2022 devant le jury composé de :

| | |
|---|---|
| **Garant** | |
| M. NOËL Thomas | Professeur, Université de Strasbourg |
| **Rapporteurs** | |
| M. DUDA Andrzej | Professeur, Grenoble INP - ENSIMAG, Grenoble |
| Mme. MITTON Nathalie | Directrice de recherche, INRIA Lille Nord Europe |
| M. VALOIS Fabrice | Professeur, INSA Lyon |
| **Examinateur** | |
| M. BEYLOT, André-Luc | Professeur, ENSEEIHT Toulouse |

# Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Julien Montavont
September 2022

## Remerciements

Les travaux présentés dans ce document ont bénéficié du soutien d'un grand nombre de personnes et je profite de l'occasion qui m'est donnée ici pour leur exprimer toute ma gratitude.

Je tenais en premier lieu à remercier vivement Thomas Noël, mon garant et collègue avec qui je travaille depuis plus de quinze années. Ta bienveillance, ton soutien et tes conseils m'ont permis de progresser dans de nombreux domaines.

Je remercie également Madame Nathalie Mitton et Messieurs André-Luc Beylot, Andrzej Duda et Fabrice Valois, membres de mon jury, pour avoir acceptés de porter leurs regards experts sur mes travaux, pour leurs critiques constructives, et pour la qualité de nos discussions à l'issue de la soutenance.

Je tiens également à remercier tous les docteurs ou futurs docteurs que j'ai eu la joie de co-encadrer durant ces dernières années : Romain, Damien, Cosmin, Sébastian, Amaury et Farzad. Les contributions scientifiques présentées dans ce document ont été enrichis par votre vision, votre inventivité et votre travail. Sans vous, ce document n'existerait pas.

Je tenais également à remercier l'équipe Réseaux du laboratoire ICube pour leur gentillesse et leur ouverture d'esprit. En particulier, merci Pierre pour ta disponibilité et le partage de ton savoir. Nos discussions et nos collaborations m'ont permis de progresser significativement dans mes pratiques pédagogiques et mes connaissances.

Enfin, je remercie chaleureusement ma famille. Merci Nelly pour ta présence à mes côtés, ta sensibilité et ton soutien. Mes journées sont plus douces en ta présence. Merci à Emma pour ta joie de vivre, ta douceur et ton ingéniosité qui illuminent mes journées. Merci à Maxine, ton tempérament passionné et ton indépendance font de toi une merveilleuse personne et me motivent à être un meilleur père. Merci à mes parents, ma sœur et mon frère pour tout et merci à Jean-Marc pour ta gentillesse et ta disponibilité.

# Abstract

Low-power and Lossy Wireless Networks (LLWN) are becoming very popular over the past few years, bringing new services to diverse areas such as healthcare, logistics, transportation, and industry. Thanks to a low energy footprint and inexpensive components, one can deploy (very) large networks to collect physical information or control systems remotely, either in buildings such as warehouses or factories or in wild environments. However, those characteristics are set against the quality of the wireless communications - bandwidth and coverage are limited, and links can be unstable and lossy. New communication protocols were therefore proposed and standardized to cope with those specific features.

LLWN are more and more deployed in scenarios involving mobile entities. Regarding its nature, an application can have strict constraints on reliability or delays that are difficult to guarantee in LLWN. Mobile entities make this task even more difficult: mobile nodes may experience disconnections, resulting in packet loss and delays. In this context, we concentrated our research effort on asynchronous LLWN and studied the mobility support at the MAC and network layers of the communication stack. Furthermore, we studied how mobile nodes can slot their transmissions into the channel access schedule while their neighbors can suddenly become unreachable. In addition, numerous scenarios involve mobile nodes as destinations, requiring specific support from the network layer. We mainly tackled this problem with two approaches: 1) using an anchor point to relay the traffic originated or destinated to mobile nodes; 2) using a routing protocol to update the routing maps towards the mobile nodes.

In parallel, asynchronous protocols are more and more abandoned in favor of synchronous protocols. The latter generally compute and allocate time/frequency blocks for each node to offer some guarantees on delay and reliability, while the former balance the latency with the energy consumption. However, mobility can cause serious synchronization issues that defeat the purpose of having a synchronous protocol. By contrast, wake-up radio, thanks to an ultra-low power consumption receiver, enable purely asynchronous communications with no tradeoff between latency and energy consumption. However, this technology comes with limitations, such as very short coverage or very limited bandwidth. We studied how we can overcome those limitations at the MAC and network layers to achieve performance on par with synchronous protocols. Yet, we do not

have a contribution to mobility support leveraging wake-up radio, but we draw some research directions in that context.

Finally, our research efforts make us consider a variety of scenarios, applications, and protocols. Even today, deploying an LLWN to meet the application needs remains a complex task: there is a plethora of protocols to choose from, and each protocol comes with numerous parameters for which the effect is not necessarily visible. In addition, the adoption of new communication protocols is relatively slow, because they generally require writing sensitive code to be run in the privileged mode of operating systems. We suggest benefiting from the emergence of Software Defined Networking (SDN) together with the cloud revolution and advances in artificial intelligence to bring more flexibility to computer networks, LLWN in particular. We propose to shift from a monolithic network stack inherited from the TCP/IP model to programmable network functions and protocols. We will address this scientific challenge with an incremental approach that first leverages SDN, cloud computing, and artificial intelligence to make autonomous networking a reality. Then, we propose to make the whole network stack programmable: every protocol can be decomposed as a list of states, transitions, and conditions that can be compiled as bytecode like any computer program. As a result, network nodes become network processors that execute a program, opening the way for ultimate flexibility, adaptability, and scalability. This challenging research project will drive our research efforts for the next years.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 General context

A wireless network is a computer network in which the connections between nodes use wireless links. Wireless networks became popular in the 2000s, with the emergence of wireless communication technologies such as Wi-Fi or 3G. Since then, wireless communication technologies drive digital communications, with mobile broadband Internet subscription growth by 11.8% per year on average in the last 10 years[1], reaching 4.2 billions (81%) of the worldwide subscriptions to the Internet in 2017 according to the International Telecommunications Union (ITU). Wireless communication technologies are a key element of ubiquitous networking, allowing users to access and exchange information at any time, from anywhere, and from any appliance, providing continuous connectivity. Ubiquitous networking contributed to the development of new application services of existing networks, including healthcare, logistics, transportation, education, and entertainment to name a few.

In the same era, advances in integrated circuits made it possible to include wireless transceivers, sensors, and actuators in one package, opening the way to network billion of everyday devices, from household electrical goods to wearable devices [1]. The networks formed by those devices are characterized by low-power consumption, low-datarate, lossy links, and short-range applications, hence the name of Low-Power and Lossy Wireless Networks (LLWN) that we will use throughout this thesis. This is a major departure from other wireless technologies that mainly focus on high datarate and long-range applications. LLWN are generally deployed over an area to collect data or take actions on the physical environment. Due to the limited radio range, the nodes rely on their neighbors to act as routers to forward the traffic further away, forming a multihop network, as illustrated in Figure 1.1. In this figure, $S$ can reach $R$ through two intermediate nodes. LLWN typically works in a converge cast scenario, in which the traffic is destinated to a single node referred to as root or sink. Routing protocols that are required to compute paths over the multihop topology generally shape the network into a tree-based structure, rooted at the sink. In addition, LLWN can

[1] https://www.oecd.org/sti/broadband/broadband-statistics-update.htm (accessed September 26, 2022)

be connected to a distribution system (e.g., Internet) via one or multiple root/gateway to enable communication with remote peers. Figure 1.1 illustrates this situation in which $S$ can communicate with the remote peer $D$ via the gateway $R$. This configuration forms what we call the Internet of Things (IoT). LLWN accounts for an increasingly huge number of connections, in which the number of connected devices will reach 125 billion by 2030 [2]. As a result, LLWN is becoming a key element of ubiquitous networking, adding values to existing applications and creating an environment that favors innovation [3].

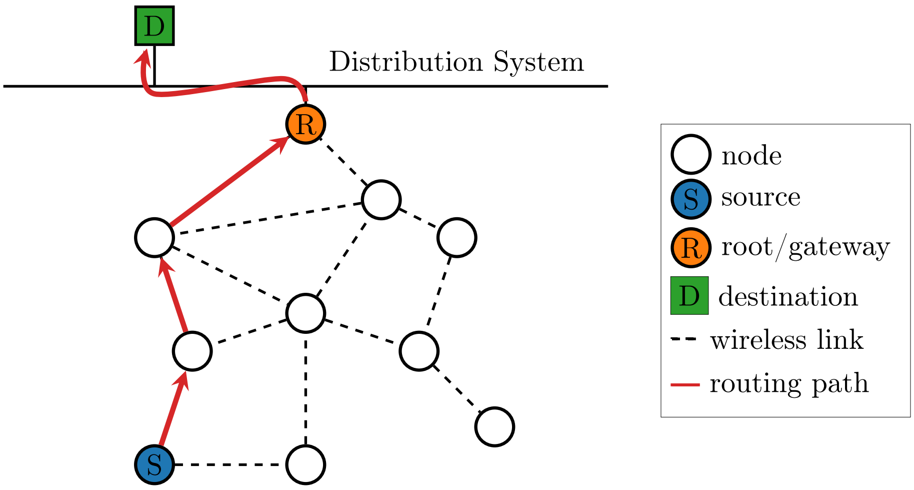



**Figure 1.1:** Example of Low-Power and Lossy Wireless Network (LLWN)

## 1.2 Motivation

In this context, mobility support, the ability to exchange information while on the move, is of crucial importance to raise ubiquitous networking to a higher level. Tracking, healthcare, intelligent transportation systems, or military are applications in which mobility support is a major concern [4]. Mobility is also a core element of various LLWN, ranging from Vehicular Ad Hoc Networks (VANET) [5] to Unmanned Aerial Vehicles (UAV) networks [6].

Unfortunately, mobility raises serious issues at different layers of the communication stack that prevent mobile nodes to communicate correctly, resulting in disconnections, packet loss, and delays in ongoing communications. The main reasons are historical because most of the paradigms and protocols still in use today were defined more than 40 years ago for wire networks which prevent any kind of mobility by nature. The wireless revolution only began in the 2000s due to the advent of technologies, such as mobile broadband, Wi-Fi, and Bluetooth. As a result, most of the communication protocols were not designed considering mobile entities. One of the most challenging tasks in LLWN is extending the network lifetime. For this, a new protocol suite has been defined by the Internet Engineering Task Force (IETF) to cope with the constraints of LLWN. This protocol suite is illustrated in Figure 1.2 and is organized as follows: application payloads are packaged by the Constrained Application Protocol (CoAP) [7] and transported using the User Datagram Protocol (UDP) [8]. Minimal security for application payloads is provided by Object Security for Constrained RESTful Environments (OSCORE) [9]. Routing functions are carried out by the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [10]. 6LoWPAN Header Compression [11] is used to compress IPv6 and UDP headers, while 6LoWPAN Routing Header (6LoRH) [12] is used to compress RPL routing artifacts present in IPv6 packets. 6top [13] is a sublayer of the Logical Link Control (LLC) that provides the abstraction of an IP link over a TSCH Medium Access Control (MAC) and schedules packets over TSCH cells. Finally, channel-access control uses the TSCH mode of 802.15.4 [14].

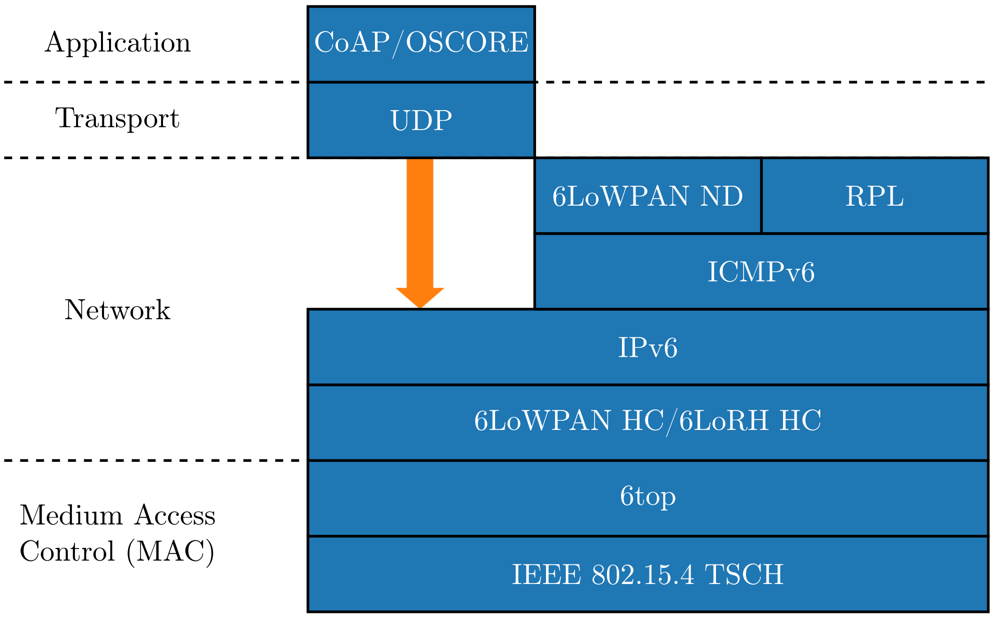



**Figure 1.2:** IETF protocol suite for LLWN

First, we can note that this protocol suite is more complex than the one for a typical network host while they are more constrained in terms of computational power, memory, and energy. To support multihop scenarios, the nodes have a dual role as router and data producer, which explains the need for extra protocols. In addition, the new 6top and 6LoWPAN sublayers add to the complexity. This contributes to making the deployment, configuration, and operation of LLWN a challenging task. I am convinced that simplification is a fundamental step toward the widespread adoption of those technologies in large-scale production deployment. This point is developed further in my future research project in Chapter 5.

At the Medium Access Control (MAC) layer, mobile nodes should slot their transmission into the channel access schedule. This operation is challenging, as the neighborhood of a mobile node keeps changing, introducing synchronization problems. Besides, the data collected and reported by mobile nodes may be of crucial importance, such as in target tracking or surveillance applications, as depicted in [15]. This topic is studied in Chapter 2. If mobile nodes only act as sources, no further actions are required. However, many applications require transmitting data to wireless nodes. Efficient downward traffic support (from a remote peer to a wireless node) is particularly challenging in LLWN [16]. Targeting mobile nodes makes the task even more difficult and requires additional mechanisms at the network layer. Three classes of solutions are possible to address this issue [17]. The first class uses a specific network entity as an anchor point to relay the traffic originated or destinated to the current logical position of a mobile node. An alternative is to involve the routing protocol in the process. Those solutions are studied in Chapter 3. The last class leverages Software Defined Networking (SDN) and is discussed in our future research project in Chapter 5.

## 1.3 Work hypothesis and methodology

The work presented in this thesis is based on the following hypothesis:

- networks are composed of fixed and mobile nodes. Future deployments are likely to use a fixed infrastructure (e.g., a distribution system) to offer global connectivity;
- networks are asynchronous. This assumption simplifies the protocol suite embedded in the nodes (e.g., 6top sublayer is removed) and the management of mobile nodes as asynchronous protocols support dynamicity by nature. Currently, synchronous networks are favored for performance reasons, but I will demonstrate in Chapter 4 that pure asynchronous protocols can outperform them in terms of delay and energy consumption in specific scenarios;
- mobile nodes are aware that they are mobiles. This assumption opens the door for solutions in which the mobile nodes can actively participate and where their traffic can be prioritized;
- mobile nodes only connect as leaves to mitigate the usage of unstable routes that increase the convergence time of the routing protocol.

My contributions were validated using network simulators, such as WSNet and Cooja. WSNet [2] is an event-driven simulator for wireless networks. It provides several radio medium models to simulate realistic link conditions. Energy consumption models are also available to compute per-node energy depletion. Then, I shifted to Cooja, a simulator for networks of ContikiOS nodes. ContikiOS is one of the most popular operating systems for resource-constrained devices. I chose this framework over other solutions because WSNet is no longer supported and to ease future experimentations because the software developed for simulations can be directly used on real sensor boards. Still, implementations and operating systems specifics have a significant impact on network performances, and such characteristics are most often ignored or simplified in simulators, and even more so in theory. I consolidated some of our results with experimentations, either over ad-hoc small-scale deployments or using large-scale experimental testbeds, such as FIT IoT-LAB. Mobility is central in my

[2] https://www.citi-lab.fr/portfolio/wsnet-simulator/ (accessed September 26, 2022)

research work, and designing reproducible experimentations that use mobile nodes remains a fastidious and time-consuming task. FIT IoT-LAB provided Cortex M3 nodes that are mobile thanks to Turtlebot2 robots. However, they are now discontinued, with no plan to bring them back online, which explains why we stick with Cooja for my last contributions.

## 1.4 Structure of the thesis

This thesis is organized into five chapters. The general introduction and main motivations are detailed in Chapter 1. Chapter 2 focus on the mobility support at the MAC layer. After a brief presentation of the associated research context, I highlight the scientific challenges of this topic and present my main contributions to tackle those challenges. Then, I draw several research directions that I think are promising. Chapter 3 moves to the network layer and details my research efforts for supporting mobility at this layer. After a brief presentation of my main contributions, I draw some conclusions and propose some potential research directions. Chapter 4 describes the wake-up radio technology and explores the benefits of using this technology at the MAC and network layers. It differs from the two previous ones in the sense that I have not yet contributed to mobility support using wake-up radios because I first needed to get familiar with this technology. Finally, Chapter 5 concludes those research activities and presents my novel research project that will drive my research efforts in the future years.

# Chapter 2

## Mobility management at the MAC layer

This chapter presents my contributions to mobility support at the MAC layer for LLWN together with research directions in this area. The work presented here is part of two Ph.D. theses.

**Supervision**

- Romain Kuntz, *Medium Access Control Facing the Dynamics of Wireless Sensor Networks*, co-supervised (33%) with Antoine Gallais and Thomas Noël (2007 - 2010)
- Damien Roth, *Mobility management in Wireless Sensor Networks*, co-supervised (50%) with Thomas Noël (2009 - 2012)

**Publications:**

- D. Roth, J. Montavont and T. Noël. *MOBINET: Mobility Management Across Different Wireless Sensor Networks*, in proc. of the IEEE Wireless Communications and Networking Conference (WCNC), 2011
- R. Kuntz, J. Montavont and T. Noël, *Improving the Medium Access in Highly Mobile Wireless Sensor Networks*, in Springer Journal of Telecommunication System, Volume 52, Issue 4, 2013

## 2.1 Research context

### 2.1.1 Introduction

In a communication protocol stack, the MAC protocol is responsible for scheduling transmissions over the wireless medium together with framing. The wireless channel is a shared medium with half-duplex capability. To avoid collisions, the MAC protocol tries to prevent concurrent transmissions in the same radio area. In addition, the most energy-consuming sub-system of a node is the transceiver module [18]. MAC protocols reduce the energy consumption of this sub-system by duty-cycling its activity. This way, a node communicates during short active periods, while its transceiver is in sleep mode most of the time. As a result, two nodes should synchronize themselves before any transmission. For this, numerous MAC protocols have been proposed in the literature [19] and can be decomposed into three categories: asynchronous, synchronous, and hybrid.

In asynchronous protocols, each node has a different wake-up time. At each wake-up, a node sample the wireless medium. If no traffic is detected, the node goes back to sleep. If the wireless channel is detected as busy, the node remains awake to receive the trailing data. This mechanism is referred to as Low Power Listening (LPL) in the literature. Before a data transmission, a source should therefore send a preamble (a carrier wave or a packet) long enough to guarantee that it will be heard by the destination [20]. Synchronized protocols require time synchronization (hence the name) between nodes to control how the wireless channel is accessed. Nodes agree on a global schedule that defines when a node can transmit, receive or sleep. However, the level of synchronization differs between protocols. For example, slotted schemes use Time Division Multiple Access (TDMA) [21] and therefore require a tight synchronization while protocols based on a common active/sleep period are more tolerant regarding time drift [22]. Finally, hybrid protocols either switch from an asynchronous strategy to a synchronous one regarding the ongoing traffic [23] or define time windows in which they alternate MAC strategies [24].

In this context, mobility consists of nodes that move while communicating. In other words, the surroundings of a mobile node keep changing, as illustrated in Figure 2.1. First, the mobile node is in the radio range of $B$ and $E$. Then, the mobile node moves in the radio range of $G$, $H$, and $F$ while $B$ and $E$ are not reachable anymore. In such a situation, the ongoing mobile node's communications should be maintained regardless of its movements.

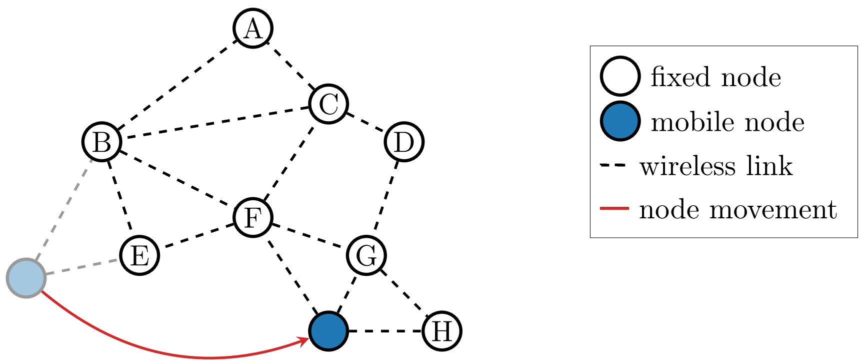



**Figure 2.1:** Mobile node moving inside a Low-Power and Lossy Wireless Network (LLWN)

First, a mobile node should slot its transmissions into the channel access schedule. Synchronized MAC protocols [21] that compute and allocate time/frequency blocks for each node seem not well suited for supporting mobile nodes due to the time required for the protocol to converge. Mobility causes synchronization loss and frequent network disruptions due to association issues encountered by mobile nodes [25]. By contrast, asynchronous MAC protocols support dynamicity by nature because there is no medium pre-reservation: a mobile node basically competes like any other node to access the channel. However, the next hop resolution remains an issue. Before transmitting a message to a remote host, a source should determine the next hop directly reachable (i.e., in its radio range) towards this destination.

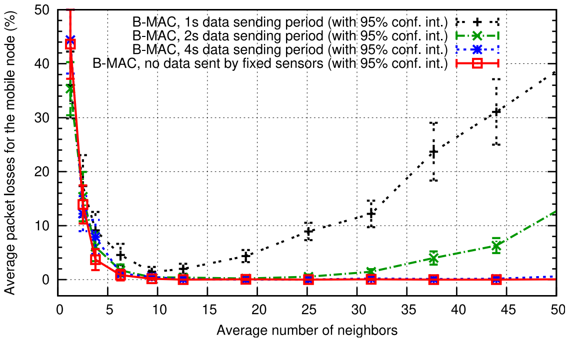
### 2.1.2 Problem statement

We consider a LLWN composed of fixed and mobile nodes using an asynchronous MAC protocol. We assume that mobile nodes act only as sources, i.e., they will never be the destination of data traffic. Before each transmission, a node should resolve the address of the direct destination. As the neighborhood of a mobile node is constantly changing, the resolution of the destination address is challenging. One easy way to tackle this problem is to force mobile nodes to broadcast their data. However, such behavior may have a significant impact on the overall network performance.

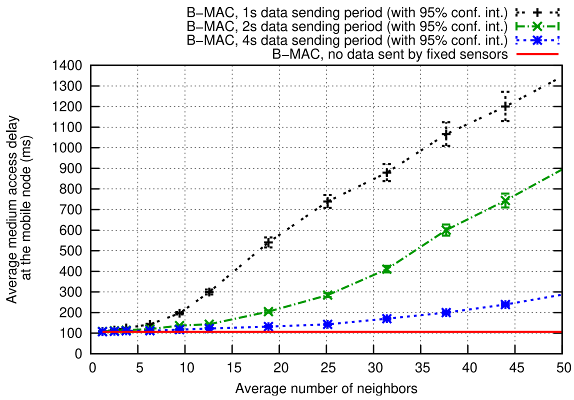
To investigate the possible shortcomings experienced by mobile nodes using a broadcast strategy with a non-optimized asynchronous MAC protocol, we performed a series of simulations in WSNet. We chose to work with B-MAC [20] because it was one of the reference preamble sampling protocols for LLWN at that time. The simulation scenario consists of a mobile node moving in a network whose density and data communication frequency of the fixed nodes are gradually increasing. Each node sends its data packet in a broadcast fashion (there is no routing) and does not forward the received packets. Therefore the broadcast scheme does not cause any exponential traffic load in the network, which remains uniform throughout the simulation. We consider a packet as lost if none of its neighbors received the broadcasted packet. Simulation parameters are further detailed in [26].

Figure 2.2 presents the average packet losses, medium access delay, and duty-cycle experienced by a mobile node regarding its average number of fixed neighbors. First, we can notice that more than 10% of the packets sent by the mobile node are lost when the average number of its neighbors is below 3.77 or above 30. Those losses are mainly the result of synchronization issues: the mobile node transmits packets while there is no reachable neighbor, or while all neighbors are sleeping because part of the preamble was not received due to non-overlapping radio ranges. Packets are also lost because of collisions — packets were not captured because the reception sensibility was better for another one, packets were successfully captured but dropped due to errors in content, etc. Those problems are directly related to broadcast transmissions that do not use acknowledgments to confirm the successful reception and prevent the usage of an Automatic Repeat reQuest (ARQ) error-control strategy. To increase the packet delivery ratio, we definitely need to enforce unicast communications by providing valid destinations to mobile nodes.

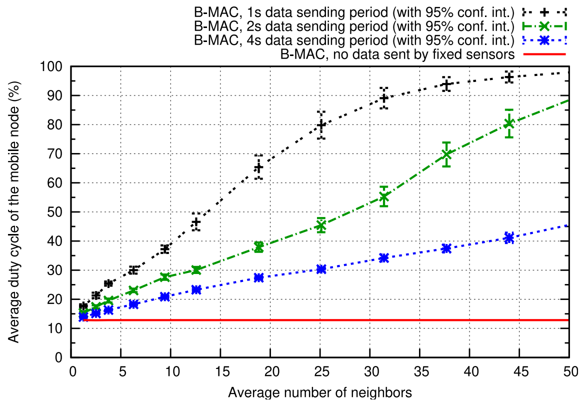
Figure 2.2b shows that the more the mobile node has neighbors, the higher the contention is,




**(a)** Average packet losses




**(b)** Average medium access delay




**(c)** Average duty-cycle

**Figure 2.2:** Network performance of a mobile sensor using B-MAC regarding to its average number of neighbors

increasing the delay to access the medium. However, if the mobile node experiences long medium access delays, its surroundings probably changed before its effective transmission, contributing to the synchronization problem. In addition, Figure 2.2c shows that the duty-cycle of the mobile node increases significantly when the competition on the medium is high. The average duty-cycle of the mobile node is 12.8% when there is no traffic from fixed nodes while it is above 30% when the contention is high, even in sparse networks. Unfortunately, a high duty-cycle increases the energy consumption. This advocate for prioritizing the medium access in favor of mobile nodes.

### 2.1.3 Challenges

Asynchronous MAC protocols have appeared as a simple solution to deal with mobility. To confirm this feeling, we performed a series of simulations to analyze the experience of a mobile node using a non-optimized broadcast strategy and identified three main issues.

First, fixed and mobile nodes experience synchronization issues, generating packet losses. Medium contention contributes to this problem by increasing the medium access delay. In addition, medium contention increases the duty-cycle, which translates in a reduced node lifetime. Finally, transmitting each packet in a broadcast fashion prevents the usage of an ARQ error-control strategy which is required to achieve a satisfactory packet delivery ratio.

To tackle those issues, we propose to:

**Contribution**

- adopt a unicast strategy by providing a valid destination address to mobile nodes. For this, the mobile nodes resolve the potential destinations by periodically listening to the wireless medium. The solution overhead is put on mobile nodes. This contribution is detailed in Section 2.2.1;
- prioritize the medium access in favor of mobile nodes. In this solution, the mobile nodes benefit from opportunistic forwarders and steal the medium acquired by fixed nodes. In contrast to the previous contribution, this solution mainly impacts the fixed nodes. This contribution is detailed in Section 2.2.2.

## 2.2 Main contributions

### 2.2.1 Resolving the destination address

**Introduction** To allow mobile nodes to use unicast, they need to resolve a valid destination address for every transmission. For this, we propose in [27] that each mobile node builds and maintains a table of potential forwarders by passively listening to the wireless medium. Active neighbors can be identified using the source or destination addresses included in the MAC header of overheard messages. The forwarder table is managed with a soft state approach to reduce the memory footprint on the mobile nodes. In this work, we focus on convergecast communication only. As a result, fixed nodes only relay packets from mobile nodes toward the gateway.

**Contribution** Forcing mobile nodes to listen to the wireless medium to overhear transmissions will increase the duty-cycle and therefore the energy consumption. To limit this phenomenon, we define four triggers for the listening process. Those triggers are presented in Table 2.1. The first one, referred to as *listen on TX*, starts the listening process whenever the mobile node has a packet to send. The mobile node turns off its radio as soon as a forwarder address is captured and the data is successfully transmitted to this node. As a result, the forwarder table is not used here. The *table empty* trigger ensures that the forwarder table has at least one active entry. When the last entry expires, the mobile node turns on its radio to look for new forwarders. The *duty-cycling* trigger periodically turns on the radio to add potential new forwarders to the table. With this trigger, if the table is empty for a pending transmission, the mobile node switches to *listen on TX* and back to *duty-cycle* after a successful transmission. Finally, *none* does not modify the duty-cycle used by the MAC protocol.

| **Trigger** | **Detail** |
|---|---|
| listen on TX | listen only if a transmission is pending |
| table empty | listen only if the forwarder table is empty |
| duty-cycling | alternate extra sleep/listen periods |
| none | keep the duty-cycle of the MAC protocol |

**Table 2.1:** Triggers for the listening process

When a transmission is pending, the mobile node should choose one entry in the forwarder table as the destination. A simple solution is to use a random strategy. However, in a convergecast scenario, a mobile node can order the table data by computing the routing maps from the overheard communications. For this, it should record the source and destination of each transmission to compute the relative path cost of each potential forwarder. Then, it can select the one that presents the lowest value. This optimization is illustrated in Figure 2.3. If the mobile node overhears a transmission between $C$ and $B$, and between $B$ and $A$, the relative path cost of $A$, $B$, $C$ are 0, 1 and 2 respectively. By transmitting its data directly to $A$ instead of $C$, the mobile node reduces by 2 the number of transmissions required to reach the gateway. This second approach is named *selective* hereafter.

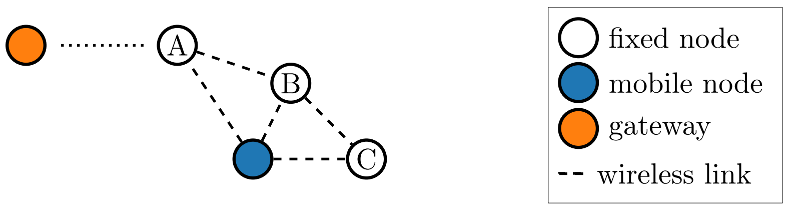



**Figure 2.3:** Minimizing the relative path cost for destination selection

**Results analysis** We implemented our triggers, listening processes, and forwarder selection in WSNet and compared the performance against a broadcast approach. Our simulation scenario involves 50 mobile nodes moving as billiard balls [28] in a 10x10 grid of fixed nodes. The mobile node traffic is set to 1pkt/4mins. All nodes use the asynchronous protocol X-MAC [29] at the MAC layer, and we emulate the consumption of a Chipcon CC1100 transceiver. Simulation parameters are further detailed in [27]. In addition to the broadcast approach, we evaluated different configurations of our triggers/listening processes. We set a specific simulation ID for each configuration, as depicted in Table 2.2. Simulation ID 1, for which the duty-cycle is dictated by the MAC protocol only, is used as a benchmark.

| Simulation ID | Trigger | Configuration | | |
|---|---|---|---|---|
| 0 | broadcast method (no listening) | | | |
| 1 | based on the MAC protocol | | | |
| 2 | duty-cycle | sleep: 1s | listen: 20ms | TTL: 10s |
| 3 | duty-cycle | sleep: 1s | listen: 100ms | TTL: 10s |
| 4 | duty-cycle | sleep: 10s | listen: 20ms | TTL: 10s |
| 5 | duty-cycle | sleep: 10s | listen: 100ms | TTL: 10s |
| 6 | duty-cycle | sleep: 60s | listen: 20ms | TTL: 10s |
| 7 | duty-cycle | sleep: 60s | listen: 100ms | TTL: 10s |
| 8 | table empty | TTL: 10s | | |
| 9 | listen on TX | | | |

**Table 2.2:** Definition of Simulation ID

Figure 2.4 represents the traffic transmitted by mobile nodes and received by the gateway. The triggers have no impact on this metric, so we grouped the results by the method used to select an entry in the forwarder table. As expected, broadcasting generates many duplicated messages (most of them during the initial transmission), but 98% of the messages sent by the mobile nodes are received by the gateway thanks to the duplicates. Our listening processes drastically limit the number of duplicated messages, but increase packet loss by 14%. The topology of the network explains the similar performance achieved by the *random method* and the *selective method*: the paths taken by the latter only differ by 2 hops at most from the ones taken by the former.

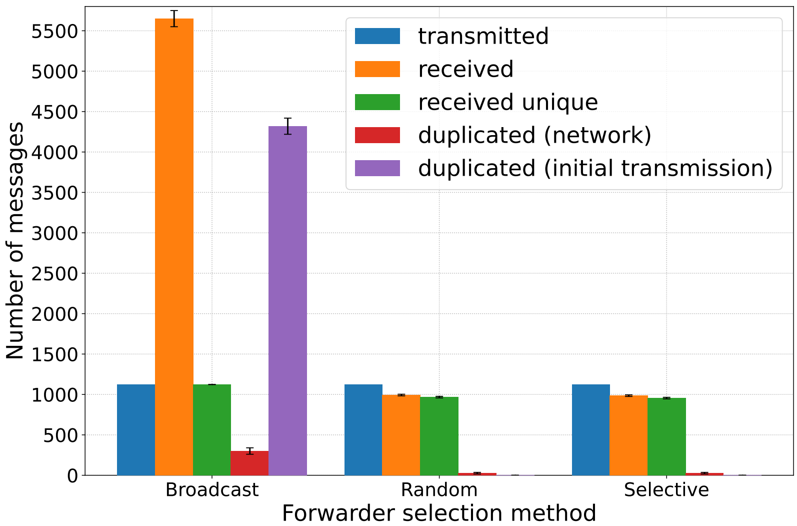



**Figure 2.4:** Traffic from mobile nodes

Broadcasting increases the energy consumption on the fixed nodes by 15% (because of duplicates), while our triggers require the mobile nodes to spend more energy to determine valid forwarders. Figure 2.5 presents the average energy consumption of a single mobile node per Simulation ID. As we can see, four of them (Simulation ID 6 to 9) consume more than twice our benchmark (Simulation ID 1). By contrast, the duty-cycle trigger with short sleep periods (Simulation ID 3 to 5) achieves lower energy consumption. This observation is clearer in Figure 2.6, which presents the percentage of messages for which a valid entry was found in the forwarder table before the effective transmission. As we can see, periodic listening processes with large sleep periods (Simulation ID 6 and 7) are inefficient: in more than 90% of transmissions the forwarder table is either empty or the entries are outdated when transmission occurs. As a result, mobile nodes are forced to listen to the channel (looking for a forwarder) before almost every transmission, causing serious energy depletion. The same observation can be made for Simulation ID 8: although a valid forwarder is set for more than 70% of transmissions, this process consumes more than the ones used in Simulation ID 2 to 5 because the radio is turned on during long periods whenever the table is empty. By contrast, frequent and short listening periods (Simulation ID 2 and 3) achieve a table hit ratio greater than 50% together with low energy consumption.

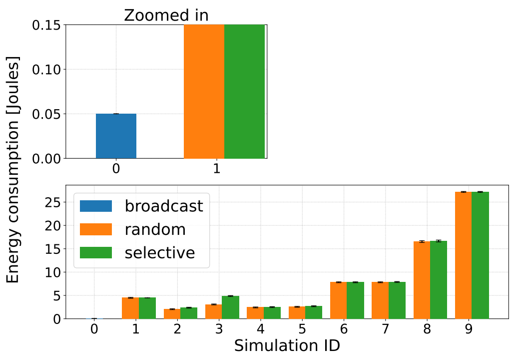



**Figure 2.5:** Energy consumption

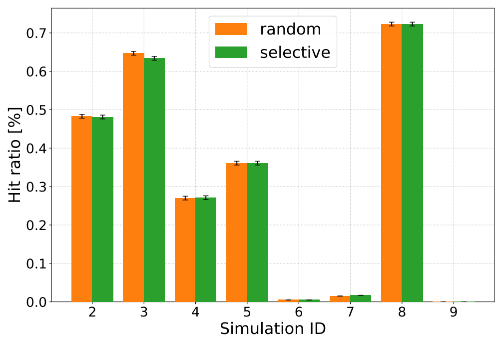



**Figure 2.6:** Forwarder table hit ratio

**Lessons learned** Our triggers and listening processes allow mobile nodes to transmit their data towards the gateway without broadcasting, limiting the overall energy consumption and therefore increasing the network lifetime. The *duty-cycling* trigger with frequent and short listening periods presents the best results in terms of energy consumption and table hit ratio. Finally, there is no significant difference between the *random* and *selective* methods to select an entry from the forwarder table but this observation should be challenged with scenarios that present dissimilar path costs between neighbor nodes.

From my point of view, this contribution is a first step towards seamless mobility but still suffers from serious synchronization problems between a mobile source and a fixed receiver. **The crux of the issue is to reduce the channel access delay for mobile nodes to ensure that a mobile source and its next hop keep synchronized during the entire transmission.** Tuning the listening processes or designing new triggers will not help in that matter, so I decided to investigate a new approach in which we can break the channel access fairness by giving more transmission opportunities to mobile nodes.

### 2.2.2 Prioritize medium access

**Introduction** To prevent mobile nodes from facing synchronization problems and experiencing long medium access delays, we propose that mobile nodes can take possession of the medium initially owned by a fixed node. In [30], we extended B-MAC [20] to add a small interval observed by fixed nodes between the preamble and the data to allow a mobile node to send its data prior to the sender of the preamble. This principle can be applied to other preamble sampling protocols, and we have carried on this work with X-MAC [29], designing X-Machiavel [31]. This protocol takes benefit of the strobed preamble mechanism of X-MAC to integrate the mobile nodes in the communication schedule. Also, we enable mobile-to-mobile communications and remove the need to maintain a list of valid forwarders, as presented in the previous section.

**Contribution** When the channel is idle, packets from the mobile node can be opportunistically forwarded by fixed nodes to the destination. When the channel is busy, the mobile node will overhear ongoing transmissions of other fixed nodes and will be able to steal the channel to send its data. These operations are possible as X-Machiavel adds *type* and *flags* fields in the packet header. In the type field, a packet will be identified as being: a preamble packet (type `P0`, `P1` or `P2`), a data packet (type `DATA`), an acknowledgment for a preamble (type `PK0` or `PK1`) or an acknowledgment for a data packet (type `ACK`). These types are presented in Table 2.3. The `flags` are used to notify the type of the packet transiting in the network. Mobile nodes always set the `M` flag, which remains set during its lifetime. This flag allows fixed nodes to forward such packets with `P2` preambles that prevent the channel from being stolen.

| **Packet type** | **Detail** |
|---|---|
| P0 | Preamble strobe from mobile node - prevents channel stealing |
| P1 | Preamble strobe from fixed node - channel is open for stealing |
| P2 | Preamble strobe from fixed node - prevents channel stealing |
| DATA | Data packet |
| PK0 | Acknowledgment of P0 from an opportunistic fixed node |
| PK1 | Acknowledgment from the intended destination |
| ACK | Acknowledgment for DATA |

**Table 2.3:** X-Machiavel packet types

A mobile node wishing to send a data packet while the medium is idle starts sending a strobed preamble of type `P0` with the `M` flag set. The destination field of each preamble packet is set to the ID of the final destination (hereinafter called *remote peer*). If that peer happens to be in the neighborhood of the mobile sensor and hears the preamble, the principles of X-MAC apply: the peer sends a PK1 acknowledgment without delays to claim the data to the mobile node. If the remote peer is not in the vicinity of the mobile node, a fixed node (hereinafter called a *forwarder*) can acknowledge it with a `PK0` acknowledgment, as illustrated in Figure 2.7. Note that only fixed nodes are allowed to act as forwarders. Upon reception, the mobile node sets the destination ID of its data packet to the source address of the received message: it explicitly destines the data to the forwarder that requests it. Finally, the forwarder acknowledges the data. The data from the mobile node is then routed through multiple fixed nodes until it reaches its final destination. Data with the `M` flag set is transported hop-by-hop toward its final destination using `P2` preambles. This prevents the channel from being stolen and potential forwarders from claiming the data.

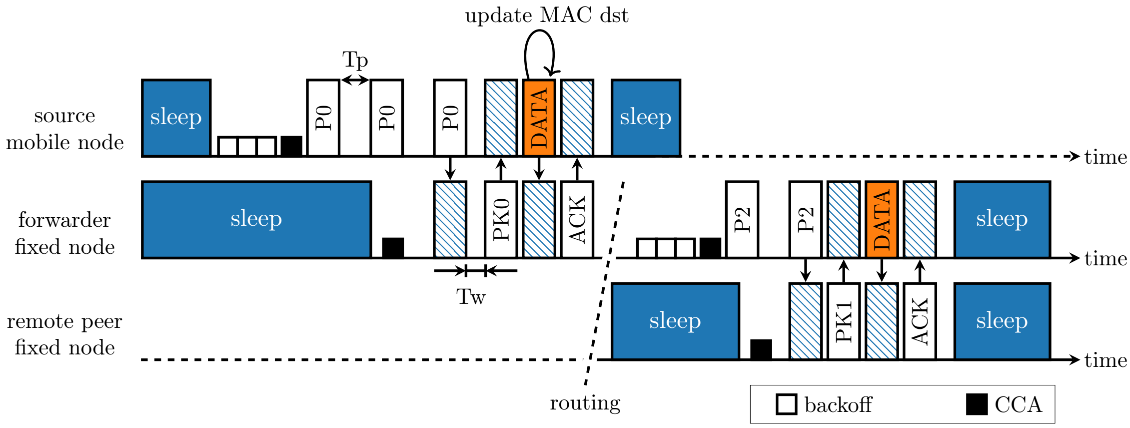



**Figure 2.7:** Mobile-to-peer communications when a forwarder acknowledges the preamble before the remote peer

We define a simple backoff mechanism to avoid collision between multiple potential forwarders. Forwarders draw a random delay $T_w$ such as:

$$T_w = random\left(\frac{T_p}{2}, T_p\right) \tag{2.1}$$

with $T_p$ being the interval between two preambles. The use of such backoff also grants priority to the remote peer. Once the $T_w$ delay expires, the forwarder performs a carrier sense and sends its `PK0`.

On an occupied channel, the mobile node can take the opportunity of the interval between strobed preambles to send its data to a forwarder, as illustrated in Figure 2.8. If a mobile node overhears a `P1` preamble, it tries to send its data to that forwarder between two consecutive preambles. After receiving a `P1` preamble, it waits for a delay equivalent to $T_w$ and sends its data to the forwarder. Before transmitting the data, the mobile node sets the destination address to the source address of the received preamble. The $T_w$ delay limits potential collisions with other mobile nodes or with the preamble acknowledgment from the real destination of the preamble. Upon reception, the forwarder still has to send its original data to its peer node (noted as *remote peer* in Figure 2.8). The forwarder sends its next preambles with a P2 type to prevent another mobile node from capturing the channel again. Overhearing this preamble type from the forwarder acts as an acknowledgment for the mobile node. If not, it can try again later upon reception of a P1 preamble. Then, the data from the mobile node is routed by the forwarder as already illustrated in Figure 2.7.

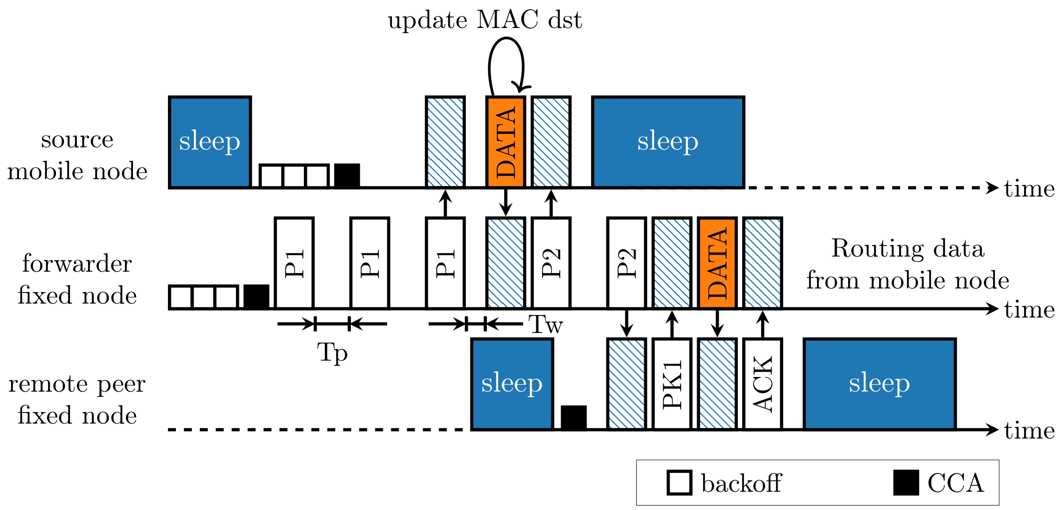



**Figure 2.8:** Mobile-to-peer communications when the channel is occupied by a fixed node

The first-hop operations previously described would mitigate synchronization and congestion issues at the mobile node. However, in a multi-hop topology, the data emitted by a mobile node and forwarded by the fixed nodes toward the final destination may still be affected by congestion. To reduce the end-to-end delay experienced by mobile nodes, X-Machiavel allows fixed nodes to steal the medium when forwarding data from a mobile node (packets with the M flag set). However, as the fixed nodes operate a routing protocol, they can only employ this mechanism when the source of the overheard preamble is the next hop computed at the routing layer.

**Results analysis** The evaluation of X-Machiavel was performed using WSNet. 16 to 96 mobile nodes are moving as a billiard ball [28] in a 4x4 grid of fixed sensors. Fixed and mobile nodes send one data packet every 10s during 1 hour, whose destination is randomly chosen among the whole set of fixed sensors. Finally, we simulate the energy consumption of a Chipcon CC1100 transceiver. Simulation parameters are further detailed in [31]. Our analysis also evaluates X-MAC to serve as a benchmark. With X-MAC, mobile nodes implement an ideal geographic routing protocol to determine the next hop for every transmission.

We focus on the results for 64 mobile nodes for readability reasons and because it is the most representative configuration. The interested reader will find the complete set of results in [32]. Figure 2.9 presents the average data packet losses at the MAC layer together with the causes of such errors. X-Machiavel reduces the packet loss by 40% compared to X-MAC. This is mainly due to the mitigation of the *no neighbor* and *data no captured* problems. The

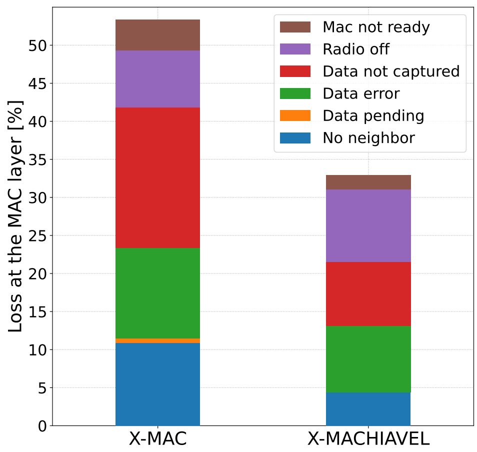



**Figure 2.9:** Average losses at the MAC layer

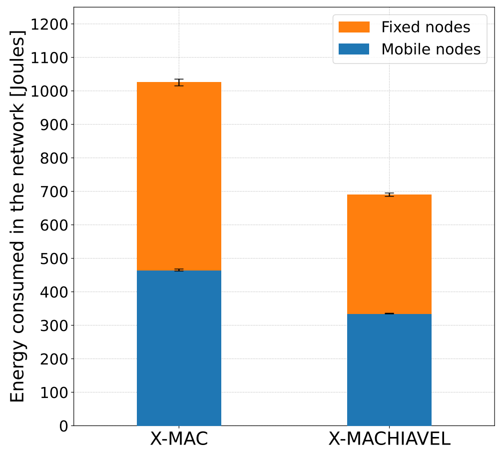



**Figure 2.10:** Energy consumed in the network at the end of the simulation

first refers to a sender and a receiver being out of sync when effective transmission occurs. In this situation, the whole strobed preamble is transmitted without being acknowledged, and the ensuing data is lost. X-Machiavel mitigates such losses because mobile nodes set the next hop upon effective data transmission. By contrast, the geographical routing protocol used with X-MAC proactively computes it, resulting in out of sync problems. *Data no captured* means that the radio has captured another packet because the sensibility was better for that packet. X-Machiavel mitigates that problem by reducing the number of concurrent transmissions. Whenever the channel is occupied, a mobile node tries to take possession of the medium instead of starting a new transmission.

Reducing the packet loss at the MAC layer has an impact on energy consumption, as illustrated in Figure 2.10. By reducing the number of packet retransmissions, X-Machiavel consumes 33% less energy than X-MAC. In addition, the savings performed on the mobile nodes which send reduced preambles or no preamble at all contribute to this reduction.

The average delay per packet to access the medium is depicted in Figure 2.11. This one-hop delay includes the initial backoff, the channel sampling period, potential congestion backoffs, and the preamble length. We can note a clear advantage for X-Machiavel: the medium access delay experienced by mobile nodes is divided by a factor of 2 when operating X-Machiavel thanks to opportunistic forwarding and medium stealing. Short medium access delays increase the medium availability, thus decreasing the congestion backoff on the fixed nodes. X-Machiavel also helps to reduce the end-to-end delay, as illustrated in Figure 2.12. Although packets can experience longer distances with X-Machiavel (a mobile node does not know whether its forwarder is closer to the destination), the end-to-end delay is drastically reduced

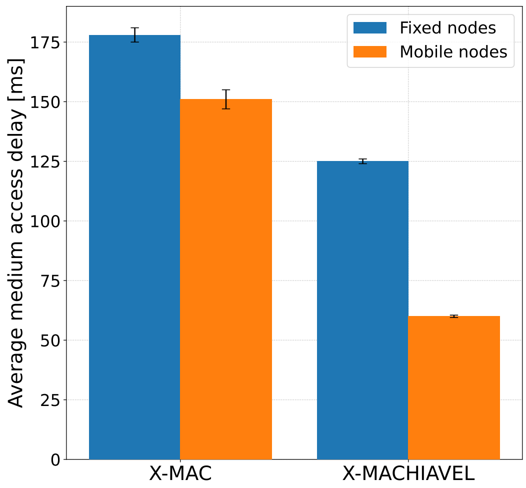



**Figure 2.11:** Average medium access delay

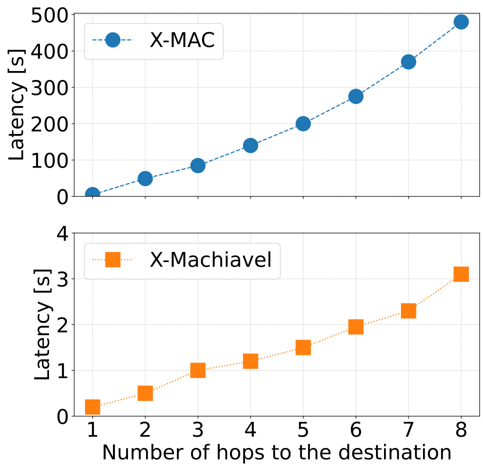



**Figure 2.12:** End-to-end delay

compared to X-MAC, which achieves 84.7s on average at 3 hops (note that there is a factor of 100 between the two Y-scales). Such large delays are mainly due to the whole set of problems experienced at the MAC layer (detailed in Figure 2.9) and multiplied exponentially with the number of hops.

**Lessons learned** By advantaging mobile nodes over fixed ones, X-Machiavel mitigates the losses at the MAC layer, which helps to reduce the medium access delay, the end-to-end delay, and the energy consumption. Network-wide, the overhead for the fixed nodes is balanced by the gain achieved on the mobile ones.

Although mobile nodes still experience packet loss with X-Machiavel, I think there is no room for improvements because we are now close to the performance limits offered by X-MAC. More modern protocols, such as ContikiMAC [33], would drive small performance increments, but **I am convinced that real breakthroughs will be made by considering new communication paradigms, such as those defined in receiver-initiated protocols or wake-up radios.** I will develop this point further in Section 2.4.

## 2.3 Conclusion

In this chapter, I presented my main achievements for supporting mobile nodes at the MAC layer in LLWN. The main problem faced by mobile nodes is to determine the address of the next hop for every communication. I showed that a simple broadcast approach is not a viable solution: it increases the medium contention together with the packet loss. In addition, it put the effort on fixed nodes, increasing the applicative traffic (due to duplication). I addressed this issue with two different approaches. Passive overhearings allow mobile nodes to maintain a list of potential forwarders that they can use whenever a transmission is scheduled. But the mobile nodes still suffer from synchronization problems, especially on a busy channel. Therefore, I guided the research to study to what extent breaking the channel fairness can limit those situations, resulting in X-Machiavel. This solution improves the packet loss, delays, and energy consumption of both fixed and mobile nodes.

Mobility support at the MAC layer for LLWN has drawn the attention of the research community over the past years, and many solutions have emerged [34]. However, only a few focus on asynchronous MAC protocols. They all consider a network composed of fixed and mobile networks, as I did. For example, MA-MAC [35] extends X-MAC with mobility detection. The received signal strength is monitored continuously to trigger a broadcast strategy when the mobile node moves away from its current forwarder. Then, the mobile node switches back to unicast when a new fixed node acknowledges the current packet. However, the experiment results show that the mobility estimation model is not reliable. From my point of view, the need for a mobility model is excessive and superfluous in asynchronous networks because switching from one forwarder to another does not involve any signaling and does not suffer from any delay. In this context, I advocate reactive strategies to tackle layer 2 mobility. In addition, it is definitely not a good idea to use broadcasting, even temporarily.

MoX-MAC [36] is similar to X-Machiavel in the sense that mobile nodes can benefit from a medium previously reserved by a fixed node. When a packet is pending and the channel is busy, a mobile node keeps its radio ON to eavesdrop on the current communication. If it captures an acknowledgment, it sends its data to the destination of this acknowledgment, which is the source of the current transmission. For this, each fixed node opens a reception window after every transmission to accept potential packets from mobile nodes. The principles of X-MAC apply if the channel is idle or if no acknowledgment is overheard. Due to the extra reception window, this approach increases the energy consumption of fixed nodes even if no mobile node is present in the vicinity. In addition, the mobile nodes still experience synchronization issues on an idle channel, as no optimization is proposed for that case.

X-MAC was proposed in 2006 and is now outperformed by other solutions, such as ContikiMAC [33]. In [37], the authors clearly show that ContikiMAC outperforms X-MAC in terms of PDR, latency, and energy consumption. In light of those results, the scientific community turns away from X-MAC in favor of ContikiMAC. M-ContikiMAC [38] focuses on burst traffic management in the presence of mobile nodes. In this protocol, a mobile node can benefit from an opportunistic forwarder similarly to X-Machiavel: the first node that acknowledges the data becomes the next hop for the pending packets of the same burst. However, multiple forwarders can receive the same data, resulting in packet duplication and potential collisions on the transmission of the corresponding acknowledgments. ME-ContikiMAC [39] resolves the duplication problem by re-introducing a preamble strobe before the burst transmission. The node that acknowledges first the preamble strobe becomes the forwarder for the pending burst. As a result, data packets are only sent in unicast, preventing duplication. Finally, MobiXplore [40] further improves ME-ContikiMAC by allowing a mobile node to directly search for new forwarders (i.e., sending new preamble strobes) when, during a burst, a data packet is not acknowledged.

## 2.4 Research directions

As we can see, there is no revolutionary solution for supporting mobility in asynchronous MAC protocol since our contributions while mobile nodes still suffer from synchronization problems. The development of Time-Slotted Channel Hopping (TSCH) for 802.15.4 [14] makes synchronous MAC protocols very popular, dragging most of the research efforts. ContikiMAC is now the *de facto* standard in asynchronous MAC protocols. Naturally, X-Machiavel can be deployed over ContikiMAC to benefit from improved performances. However, I think such a direction is a dead end: in asynchronous protocols, the local channel contention dictates the overall performance, and favoring further the mobile nodes should degrade the observed performance. Also, such approaches prevent us from relaxing the need for a fixed infrastructure. I am convinced that real breakthroughs can come from shifting the communication paradigm.

In this context, I find interesting to investigate the mobility support in receiver-initiated MAC protocols [41]. To the best of my knowledge, this not yet studied in the literature. Introduced in 2004 with RICER [42] and popularized by RI-MAC [43], those protocols allow receivers to be the initiators of every communication by regularly wake up and announce their availability by broadcasting a beacon. When a source has a packet pending, it turns ON its radio and waits for the beacon of its intended receiver. Upon reception, the source transmits its data and gets back an acknowledgment. Acknowledgments also serve as extra beacon, advertising a new receiving window for that very receiver. By contrast to sender-initiated protocols, receiver-initiated schemes reduce the channel contention and overhearings. Such paradigm can be extended to support mobile nodes. Indeed, mobile nodes can resolve the destination address upon beacon reception. As a result, the delay between the address resolution and the effective transmission should be drastically reduced, mitigating the synchronization problems.

However, receiver-initiated protocols suffer from long idle listening periods that can be the cause of serious energy depletion [41]. In a way, our *Listen on TX* trigger (cf. Section 2.2.1) presents a similar approach and achieves high energy consumption, but the duration of the idle listening is traffic dependent. Fortunately, optimizations can be applied to reduce the idle listening. For example, senders can strobe CCA until activity on the channel is detected [44]. In a mobile node scenario, the idle listening should be limited as mobile nodes do not wait for a specific beacon, but for any beacon. Still, mitigating collisions between multiple receivers and broadcasting support remain challenging with this paradigm. Multi-channel support is an interesting direction for addressing those challenges [45].

A second interesting research direction is to benefit from wake-up radio to design a pure asynchronous MAC protocol with mobility support. Thanks to its ultra low-power consumption, a wake-up receiver can be always ON, ready to wake up (hence the name) the main radio for a pending communication. Furthermore, this technology introduces an extra communication channel that can be used for the control plane, opening the way for new asynchronous protocols freed from the limitations of the existing ones. I have already started studying the pros and cons of this technology, and my first contributions are detailed in Chapter 4.

In this chapter, I assumed that mobile nodes act as sources only because packet reception involves the network layer for path computation. However, computing routes over LLWN where mobile nodes bring dynamicity is challenging. In the next chapter, I study how mobility impacts the network layer and present several contributions that tackle the raised issues.

# Chapter 3

## Mobility management at the network layer

This chapter presents my contributions to mobility support at the network layer for LLWN together with research directions in this area. The work presented here is part of two Ph.D. theses.

**Supervision**

- Damien Roth, *Mobility management in Wireless Sensor Networks*, co-supervised (50%) with Thomas Noël (2009 - 2012)
- Cosmin Cobârzan, *Internet of Highly Mobile Things*, co-supervised (50%) with Thomas Noël (2012 - 2015)

**Publications:**

- R. Roth, J. Montavont and T. Noël, *Performance Evaluation of Mobile IPv6 Over 6LoWPAN*, in proc. of the ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PEWASUN), 2012
- J. Montavont, D. Roth and T. Noël, *Mobile IPv6 in Internet of Things: Analysis, Experimentations and Optimizations*, in Elsevier Ad Hoc Networks Journal, Volume 14, 2014
- C. Cobârzan, J. Montavont and T. Noël, *Analysis and Performance Evaluation of RPL under Mobility*, in proc. of the IEEE Symposium on Computers and Communications (ISCC), 2014
- C. Cobârzan, J. Montavont and T. Noël, *Integrating Mobility in RPL*, in proc. of the International Conference on Embedded Wireless Systems and Networks (EWSN), 2015
- C. Cobârzan, J. Montavont and T. Noël, *MT-RPL: a cross-layer approach for mobility support in RPL*, in EAI Endorsed Transactions on Internet of Things, Volume 2, Issue 5, 2016

## 3.1 Research context

### 3.1.1 Introduction

In the previous chapter, we investigated the issues related to mobility at the MAC layer and presented two main contributions to improve the quality of communications in the presence of mobile nodes. However, we only considered mobile nodes operating as sources in those works. Many applications also require transmitting data to wireless nodes, being mobile or not, for maintenance/configuration, actuation, data querying, etc. Industrial environments such as automation systems or Intelligent Transportation Systems (ITS) have an increasing demand for connectivity and mobility between nodes [46], [47]. In addition, LLWN can be provided with an IP connectivity by the mean of the 6LoWPAN adaptation layer [48]. By pushing IP technology to LLWN, one can benefit from more than 40 years of experience, smooth interoperability, and provides many tools for network diagnosis, supervision, and operation. IP connectivity also opens up a wide range of applications, including bidirectional communication scenarios.

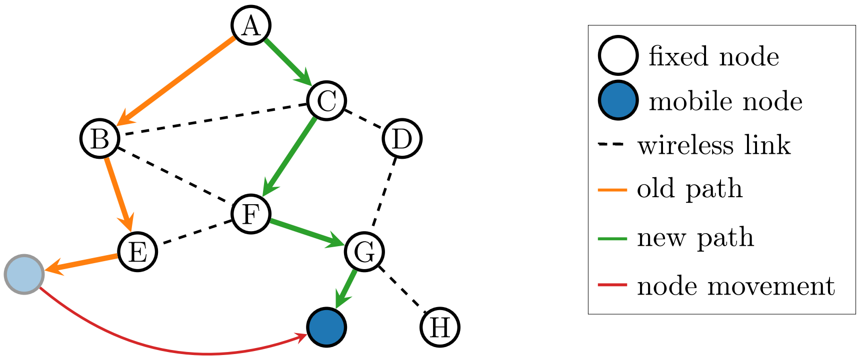



**Figure 3.1:** Delivering packets to a mobile node moving inside a Low-Power and Lossy Wireless Network (LLWN)

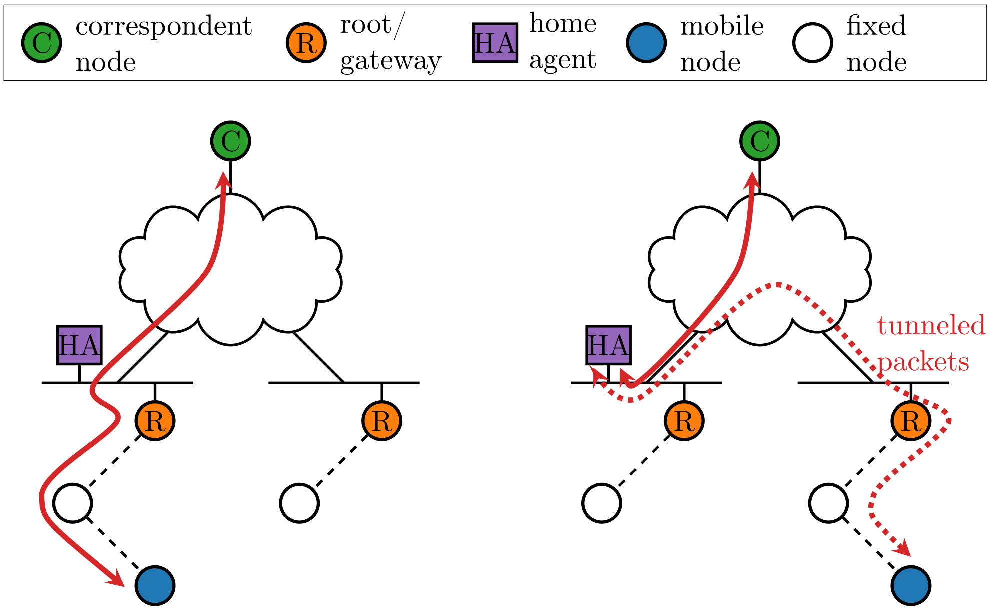
Delivering packets to a wireless node requires to know its network location. This information can be retrieved from the node address if a single namespace is used to express identity and location (as in IP), or by providing routing directions on intermediate nodes. Figure 3.1 illustrates a situation in which $A$ would transmit packets to a mobile node. In the beginning, the mobile node connects to $E$ and obtains an address that belongs to $E$, or specific routing directions to reach the mobile node are distributed among the nodes. In the former case, $A$ can deduce that the mobile node is connected to $E$ from its address and look up in its forwarding table how to reach $E$. In the latter case, $A$, $B$, and $E$ have an entry in their forwarding tables to join the mobile node. When the mobile node moves from $E$ to $G$, it updates its address accordingly, or new routing directions are spread to the network before packets from $A$ can reach the mobile node again. While the latter is handled by the routing protocol that may converge slowly, the former may break ongoing communications as the destination address can be part of the session identifier. Also, updating the destination address on $A$ requires specific support.

Mobile IPv6 [49] is the IETF standard to manage IPv6 mobility and uses native IPv6 features such as header extension and IP encapsulation that remain compliant with 6LoWPAN. However, the packets generally exchanged over LLWN have a relatively limited size (e.g., 127 bytes in IEEE 802.15.4). Packet encapsulation used by Mobile IPv6 reduces the size left for data and thus may generate fragmentation. This contributes to consider Mobile IPv6 as a non-practical solution for network layer mobility support in LLWN by several studies [50], [51]. However, the lack of information about numerous parameters makes those experimentations unreproducible. In addition, we will study how the routing protocol can handle mobility as an alternative to solutions based on a relay agent, such as Mobile IPv6. For this, we will consider the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [10], which is one of the most popular routing protocols in LLWN.

### 3.1.2 Problem statement

Mobile IPv6 uses a relay agent (named *home agent* thereafter) to redirect IPv6 packets destined to or originated from mobile nodes. Mobile IPv6 segments the Internet in two types of networks: home and visited. The home network is the administrative network of a mobile node and includes the home agent. While in the home network (*at home*), a mobile node uses its *home address* and would communicate with its peers according to the standard IPv6 protocol, as illustrated in Figure 3.2 (a). Once it connects to a visited network (*away from home*), the mobile node would acquire a new locally valid *care-of address* using conventional IPv6 mechanisms, such as stateless or stateful auto-configuration. To maintain ongoing sessions, the mobile node provides this address to its home agent to enable traffic relay through a bidirectional IPv6-in-IPv6 tunnel, as illustrated in Figure 3.2 (b). By this means, the mobile node remains reachable by its correspondents via its home address.

The performance of Mobile IPv6 is closely tied up with the reactivity of the movement detection (i.e., the detection of the arrival in a visited network), as already identified in traditional networks [52]. Mobile IPv6 relies on router advertisement for movement detection, but they are transmitted in unicast upon solicitation in LLWN [53]. A mobile node can use one of the various lifetimes defined in neighbor discovery [53] to trigger the transmission of new router advertisements. Before the shortest lifetime expires, a mobile node sends a unicast router solicitation to its current router. After multiple unsuccessful attempts, the router is considered as unreachable, and the mobile node starts looking for a new router by transmitting multicast router solicitations. After receiving a new router advertisement, the mobile node creates a new care-of address and registers this address to its new router. Upon successful registration, the mobile node finally sends a binding update to the home agent and waits for its acknowledgment. Unfortunately, the smallest maximal lifetime value is the default router lifetime set to 18 hours [54] (versus 45.5 days for the maximum address registration lifetime). As a result, mobile nodes may experience very long disconnections,




(a) Mobile node in the home network (b) Mobile node in a visited network

**Figure 3.2:** Mobile IPv6 operations

disrupting ongoing applications.

The routing protocol can also handle mobility. Routing packets in LLWN can be achieved by the IETF IPv6 routing protocol for Low-Power and Lossy Networks (RPL) [10]. This protocol builds a Destination Orientated Directed Acyclic Graph (DODAG) shaped by a set of metrics/constraints. A node connects to the graph when it receives a DODAG Information Object (DIO) from a neighbor node. Upon reception, the node sets the source as its parent (the next hop towards the root of the graph) and computes its rank (an estimation of its relative position in the graph). If multiple DIO are received, the node chooses the *best* parent regarding the computed ranks. If a mobile node connects to the graph, it can be later on disconnected from its current parent while on the move. However, RPL does not define mechanisms to detect the unreachability of a parent. This is the same problem as for the movement detection in Mobile IPv6. In short, a (mobile) node only changes its current parent upon reception of DIO with a better rank, or if a new version of the DODAG is built. However, the ranks are stable with certain metrics (e.g., Hop Count). As a result, a mobile node may stick with an unreachable parent because all available neighbors present a rank higher than its current rank, as illustrated in Figure 3.3. The rank of the mobile node is set to 3 because the rank of its parent ($E$) is set to 2. When moving away from $E$, the mobile node should select a new parent between $G$ and $H$, increasing its rank to 4 or 5 respectively, which is not allowed by RPL. As a result, the mobile node discards the DIO from $G$ and $H$ and remains virtually connected to $E$.

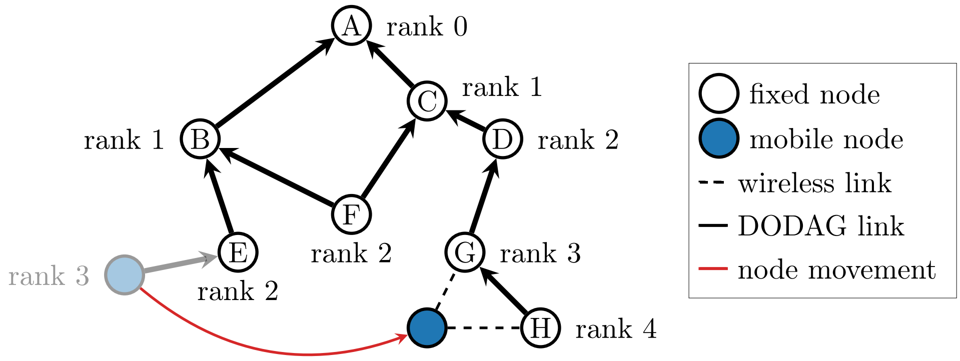



**Figure 3.3:** Mobile node entering in an area with higher ranked neighbors

To investigate further those situations, we performed experimentations and simulations for Mobile IPv6 and RPL respectively. We implemented Mobile IPv6 and updated the implementation of Neighbor Discovery to comply with the adapted version for LLWN [53]. During the experiment, a mobile node moves across two visited networks, being at one hop of the gateway in each case. For the movement detection, we use the expiration of the default router lifetime and we set its value to 30 sec. to speed up the experiments. In addition, we reduce the delay between two consecutive retransmissions of router solicitations from 10 sec. to 1 sec. For RPL, we used the WSNet simulator over a grid topology composed of 36 fixed nodes and 5 mobile nodes moving as billiard balls [28]. RPL is configured in non-storing mode using Hop Count as the metric. Experimentation and simulation parameters are further detailed in [55] and [56] respectively.

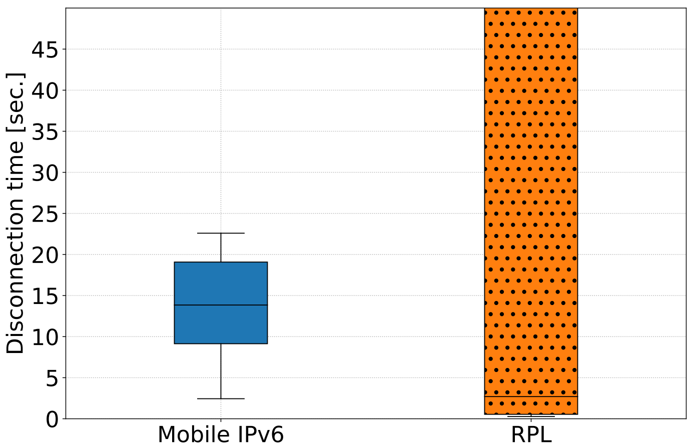



**Figure 3.4:** Disconnection times with Mobile IPv6 and RPL

Figure 3.4 represents the disconnection times experienced by the mobile nodes in Mobile IPv6 and RPL. In Mobile IPv6, the handover takes about 13.7 sec. on average to complete. The large range is related to the time at which the mobile node moves because the delay to confirm the router reachability varies between 0 and 30 sec. After the movement detection, the Mobile IPv6 operations take approximately 130 ms to complete. Here, we confirm again that the performance of Mobile IPv6 is closely related to the reactivity of the movement detection procedure. To save more energy, we can extend the default router lifetime to its maximum value, but mobile nodes may experience very long disconnection times (up to 18 hours [54]) that may significantly impact the application.

With RPL, the disconnection times can rise to 700 sec. in the worst cases, as the mobile nodes only change their parents upon the reception of a DIO including a lower rank. Those results were obtained with a very stable metric and can be mitigated with dynamic metrics, such as the Expected Transmission Count (ETX) [57]. With the connection loss, the quality of the link between the mobile node and its parent measured by ETX will decrease slowly, while the one with potential parents will increase, allowing the mobile node to change its parent. Still, such metrics may converge slowly, and they directly depend on the traffic.

### 3.1.3 Challenges

Enabling downward traffic to mobile nodes requires knowing their current network location. For this, the mobile nodes can update a relay agent to remain reachable via a single address whatever their locations are, as in Mobile IPv6. Alternatively, they can update the routing maps whenever they move. Our experimentations show that Mobile IPv6 is usable over LLWN, but the movement detection as defined in the standard takes too much time to complete (up to 18 hours in the worst case). Such long disconnection can not be distinguished from a failure of a mobile node. Similarly, our simulations show that RPL also presents long disconnection times (up to 700 sec.) with a stable metric. Even with a more dynamic metric such as ETX, RPL may converge slowly and still present serious connection loss. To tackle those issues, we decided to focus our work on movement detection in Mobile IPv6 and RPL. More precisely, we propose to:

**Contribution**

- Extend our work presented in Section 2.2.1 and use the neighborhood of a mobile node to detect the arrival in a visited network in Mobile IPv6. This contribution is detailed in Section 3.2.1;
- Use a reverse trickle algorithm in RPL to allow mobile nodes to monitor the reachability with their current parent. This contribution is detailed in Section 3.2.2;
- Extend our work presented in Section 2.2.2 to trigger an update of the RPL routing tables whenever a mobile node benefits from an opportunistic forwarder or successfully steals the medium. This cross-layer scheme is presented and compared to additional unreachability detection mechanisms suggested by RPL in Section 3.2.3.

## 3.2 Main contributions

### 3.2.1 Improving Mobile IPv6 movement detection

**Introduction** Mobile IPv6 suffers from various limitations such as the protocol's poor support (or lack thereof) for rapid and seamless handovers, mainly because of the movement detection. To tackle this issue, we propose to extend our previous work on passive overhearings to improve this mechanism [58]. Whenever the neighbors of a mobile node change, it will send a new router solicitation to trigger the reception of fresh router advertisements. This solution is generic enough to be used with a large set of MAC protocols because it does not rely on specific mechanisms defined at the MAC layer. This contribution is referred to as *Mobinet* in the following.

**Contribution** To keep track of their neighbors, each mobile node manages a table of potential forwarders. Once a mobile node overhears a packet, it records the layer 2 source and destination addresses in its table. The table works in a soft state mode - an entry for which the lifetime expires is removed. Detecting changes in the forwarder table on a periodic basis is achieved through checksum computation. Although a hash function (such as MD5) is likely to be more robust to collisions, we implemented a checksum for its simplicity and its ability to provide the same result if the order of data entries change. At short intervals, the mobile node computes a checksum over all entries in the table and compares the returned value with the previous one. If the checksums differ (the table has been modified), a counter is incremented. Whenever this counter is equal to a certain threshold, referred to as *mobility threshold*, the mobile node sends a multicast router solicitation to request the transmission of fresh router advertisements. Transmitting the router solicitation directly in multicast allows the mobile node to reach a router, being its current one or new ones. The reception of a new IPv6 prefix triggers the Mobile IPv6 handover: the mobile node creates a new care-of address and updates its home agent.

Mobinet could increase the energy consumption due to overhearing and message processing, as investigated in Section 2.2.1. However, using the duty-cycle trigger (see table 2.1) with relatively short sleeping periods (ranging from 1 to 10 sec.) limits the energy consumption.

**Results analysis** We extended our previous implementation of Mobile IPv6 for Contiki with our new movement detection. The experimental scenario is almost the same, but we include various fixed nodes (ranging from 0 to 7) in each visited network. They are organized in a line to increase (or reduce) the number of hops between the mobile node and the router. A static mesh-under routing protocol is used on fixed nodes to forward the traffic. Each fixed node transmits between 20 and 36 bytes of data every second. The listening process is set to 1 sec. of sleep for 20 ms of listening. This corresponds to *Simulation ID 2* in Table 2.2 for which we obtained a good tradeoff between energy consumption and table hit ratio. The mobility threshold is set to 1 to improve the reactivity. In addition to the evaluation of

Mobinet, we analyze the impact of 6LoWPAN compression on signaling and data traffic. We have deployed three Mobile IPv6 firmwares with different compression levels, as depicted in Table 3.1. Note that Mobinet uses full compression. Experimentation parameters are further detailed in [55] and [58].

| Firmware | Compression |
|---|---|
| none | no compression, plain IPv6 and Mobile IPv6 headers |
| ipv6 | compress IPv6 headers with IPHC [11] |
| full | compress IPv6 headers with IPHC [11] |
| | and Mobile IPv6 extensions, as defined in [59] |

**Table 3.1:** Firmwares deployed in the experimentations

The results presented in Figure 3.5 show the disconnection time related to a mobile node moving in a new network. As we can see, Mobinet allows the mobile node to recover from the handover in 1.5 sec. on average. In our scenario, overhearing a single transmission in the new visited network is sufficient for reaching the mobility threshold (a new table entry generates a new checksum value which increases the number of changes by 1) and triggers the transmission of a multicast router advertisement. As the fixed nodes transmit data every second, Mobinet is very reactive. However, this delay is closely related to the mobility threshold and traffic. Unfortunately, there is no ideal static value for the mobility threshold, as diverse datarates can coexist. The precise evaluation of a dynamic adaptation of the mobility threshold still needs to be investigated. Compressing the headers have a minor impact for Mobile IPv6, but still reduces the disconnection time by almost 1 sec. As a result, the disconnection time achieved by Mobinet in this configuration may double without compression.

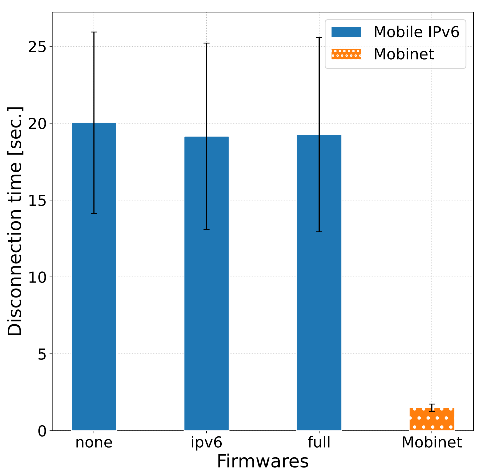



**Figure 3.5:** Average disconnection time

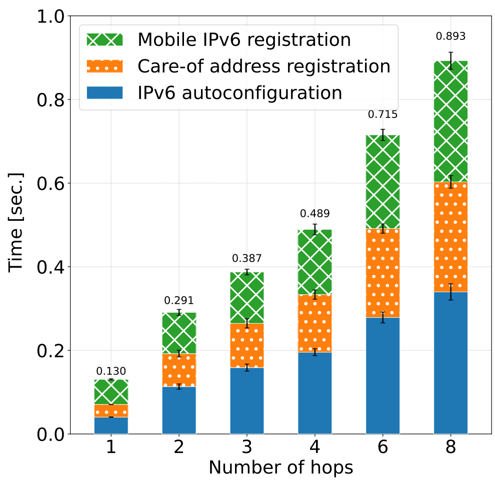



**Figure 3.6:** Duration of Mobile IPv6 op.

After the movement detection, the Mobile IPv6 operations take approximately 130 ms to complete as we can see on Figure 3.6. We chose to only report values for the *full* firmware as the results for other firmwares present very similar results. Also, Mobinet is not involved in that part of the procedure, so the results between Mobile IPv6 and Mobinet are identical. Increasing the number of hops between the mobile node and the router of the visiting network increases linearly the time required to complete the Mobile IPv6 operations, each hop adding 150 ms on average to reach a maximum value close to 900 ms for 8 hops. Nevertheless, the duration of the overall Mobile IPv6 procedure (after the movement detection) remains below 1 sec. for 8 hops.

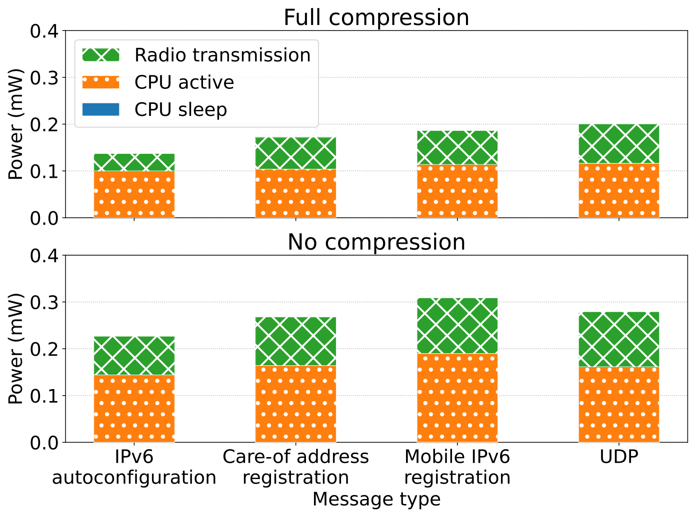



**Figure 3.7:** Mobile node power consumption

Finally, compressing the headers allows the mobile node to spend less energy, as illustrated in Figure 3.7. Those results were obtained thanks to the Energest library of Contiki. They present the power consumed for each stage of the Mobile IPv6 operations and for transmitting a data packet (UDP encapsulated in an IPv6-to-IPv6 tunnel). The compression reduces the size of the transmitted bytes, reducing the duration of the transmission and the usage of the CPU.

**Lessons learned** We proved that Mobile IPv6 is a practical solution for network mobility support in LLWN. Compression mechanisms reduce the transmission duration of tunneled IPv6 packets and Mobile IPv6 signaling together with energy consumption. We also extended our work on passive overhearings to improve movement detection and drastically reduced the duration of disconnections.

However, I think this solution suffers from several limitations. Identifying new neighbors or removing old ones triggers the transmission of new router solicitations, even if the mobile node remains connected to the same neighbor. This generates unnecessary signaling that contributes to energy depletion. For me, **an efficient solution should include MAC**

**layer events in the equation to reflect the current status of the mobile node.** More generally, solutions based on a relay station introduce a single point of failure and require solid authentication mechanisms to prevent traffic from being rerouted by malicious nodes. Finally, using a relay station introduces a triangular routing that can lead to sub-optimal traffic paths. For all those reasons, I gave up this research direction and proposed to focus on involving the routing protocol in mobility support, the nodes already assuming the role of routers.

### 3.2.2 Integrating mobility in RPL

**Introduction** Routing packets in LLWN can be achieved by the IETF IPv6 routing protocol for Low-Power and Lossy Networks (RPL) [10]. This protocol builds a Destination Orientated Directed Acyclic Graph (DODAG) using a gradient-based approach. Mobile nodes should connect as leaves to limit their impact on the network connectivity and stability [60]. When a mobile node moves and changes its parent, it can send a Destination Advertisement Object (DAO) towards the root to update the routing maps accordingly. However, the performance of the movement detection in RPL is dependent on the metric used, as overviewed in Section 3.1.2. A mobile node would use the DIO from its parent to monitor its connectivity to the graph, but they are transmitted using a trickle timer [61]. As a result, the periodicity of DIO is directly related to the stability of the network and is not easily predictable. In addition, a mobile node may wait for a long period (up to 2.3 hours [62]) before receiving the next scheduled DIO due to the trickle algorithm. To tackle those issues, we propose to extend RPL with a reverse trickle algorithm to allow mobile nodes to monitor the connectivity with their current parent [63].

**Contribution** The standard trickle algorithm starts with short intervals between two consecutive DIO to quickly react to topology changes and increases the intervals when the network is stable to limit the control traffic overhead. To support mobile nodes, we propose a reverse trickle algorithm that starts from a pre-defined maximum value and halves the sending intervals after each new DIO. As a result, the mobile nodes can compute the schedule for the next DIO and react when multiple DIO are missed in the envisioned time window.

A parent switches from the standard trickle timer to the reverse one whenever a mobile node connects to it. This is triggered by the reception of a DAO with a new *Mobility Flag* (MF) set, as illustrated in Figure 3.8. The MF flag limits the usage of the reverse trickle algorithm to areas where mobile nodes are present without affecting the rest of the network. The fixed node sends its first DIO after $I_0 \leq I_{max}$ and schedules the next DIO for $I_1 \leq \frac{I_{max}}{2}$. When the minimum value for the reverse trickle timer is reached (i.e., $I_n \geq I_{min}$), the parent of the mobile node sends a multicast DIO with an increased Destination Advertisement Trigger Sequence Number (DTSN) to force any attached node to send back a fresh DAO. If the parent receives at least one DAO with the MF flag set, it resets the reverse trickle timer to

$I_0$ and restarts the algorithm. Else, it switches back to the standard trickle algorithm to save energy.

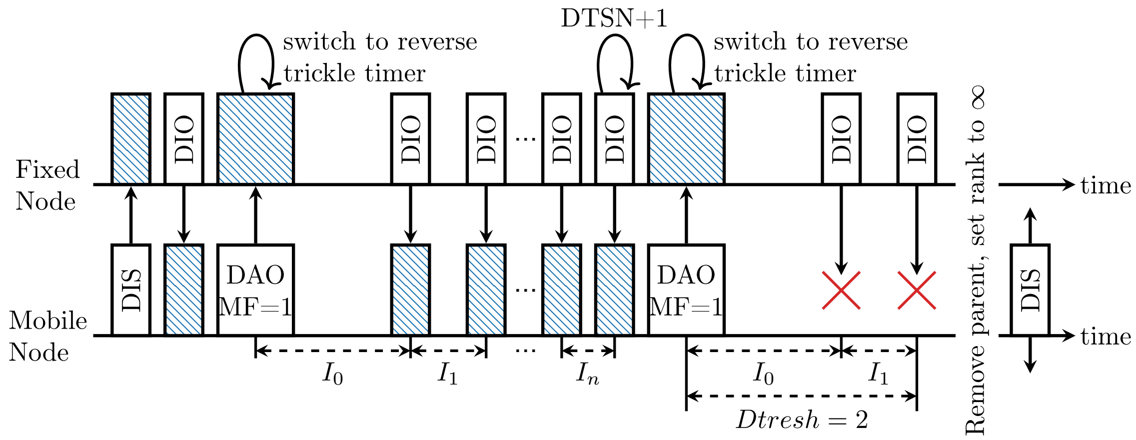



**Figure 3.8:** A mobile node connecting to a DODAG, triggering the reverse trickle timer

Mobile nodes monitor the time since the last received DIO from their parents to quickly react to a connection loss. Let *Dthresh* denotes the sum of intervals between consecutive DIO in the reverse trickle algorithm. For example, $Dthresh = 2$ means that we sum $I_0$ and $I_1$. Upon DIO reception, each mobile node sets a timer to a value corresponding to *Dtresh* in ms. If the mobile node misses DIO, this timer will expire, triggering a parent change: the mobile node removes the parent, sets its rank to infinite, and searches for a new parent by sending DIS, as illustrated in Figure 3.8. In other words, the reverse trickle algorithm gives mobile nodes a hint when the next DIO will be transmitted while *Dthresh* balances reactivity with energy savings. The maximum value for *Dthresh* is the sum of the maximum number of DIO intervals that can be obtained between $I_{max}$ and $I_{min}$.

**Results analysis** We implemented this proposal using WSNet and compared it with two solutions from the literature referred to as PeriodicDIO [64] and DynamicDIS [60]. The first removes the trickle algorithm for DIO and replaces it with a fixed value ranging from 2 sec. to 10 sec. The second proposes a dynamic DIS management procedure. In a nutshell, mobile nodes send DIS at different intervals to force the refresh of routing information. They define two thresholds referred to as *DownDIS* and *UpDIS*, which are the number of parents change and the number of times a mobile node stays connected to its parent respectively. If *DownDIS* is crossed, the inter-DIS interval should be divided by 2. Conversely, if *UpDIS* is crossed, the inter-DIS interval is multiplied by 2. Finally, $I_{min}$ and $I_{max}$ are the minimum and maximum time intervals for the inter-DIS. Those solutions use RPL internals to switch from one parent to another: upon DIO reception, the mobile node changes its parent if the computed rank is better than the current one. As overviewed in Section 3.1.2, this requires configuring RPL with a dynamic metric, such as ETX, to mitigate the disconnection time.

We simulated 100 fixed nodes organized as a square grid. The DODAG root is located in the middle of the grid. Ten mobile nodes are distributed over the grid and move randomly inside the area covered by the fixed nodes. All the nodes send data to the root at either 1 pkt/10s (for fixed nodes) or 1 pkt/s (for mobile nodes). We also set different configurations for the analyzed schemes (reported in Table 3.2). For PeriodicDIO and DynamicDIS we took the recommended values. To provide a fair comparison with those solutions, we configured RPL with ETX. Simulation parameters are further detailed in [63].

| Scheme | Parameter |
|---|---|
| Reverse-trickle 1 | $I_{min1} = 2^{10}$ ms and $I_{max1} = 2^{24}$ ms, Dthresh = 1, 3 and max |
| Reverse-trickle 2 | $I_{min2} = 2^{11}$ ms and $I_{max2} = 2^{25}$ ms, Dthresh = 1, 3 and max |
| PeriodicDIO | 1 DIO/2s and 1 DIO/10s |
| DynamicDIS | $I_{min} = 3$ sec., $I_{max} = 60$ sec., $I_{init} = 3$ sec. or 30 sec.<br>DownDIS = 1, UpDIS = 5 |

**Table 3.2:** Simulation parameters for the analyzed schemes

Figure 3.9 represents the disconnection time experienced by mobile nodes when they lose the connection with their preferred parents. DynamicDIS and PeriodicDIO rely only on RPL to change their preferred parent and thus can take a long time before reconnecting to the graph (up to 102 sec. for PeriodicDIO). Mobile nodes need to receive numerous DIO from new parents for ETX to converge and present better ranks than the current one, even if the link quality with the preferred parent continuously decreases. By contrast, our solution allows mobile nodes to change their parents independently of the metric in use. A mobile node disconnects from the graph and multicasts DIS whenever *Dthresh* expires. Any fresh DIO allows the mobile node to reconnect to the graph in this situation. As a result, the disconnection time is bounded by *Dthresh*.

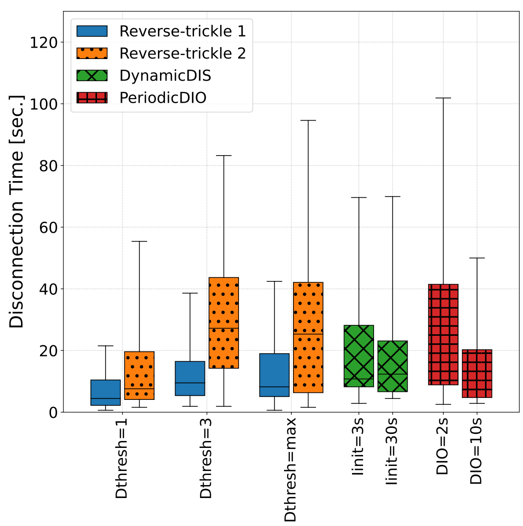



**Figure 3.9:** Disconnection time

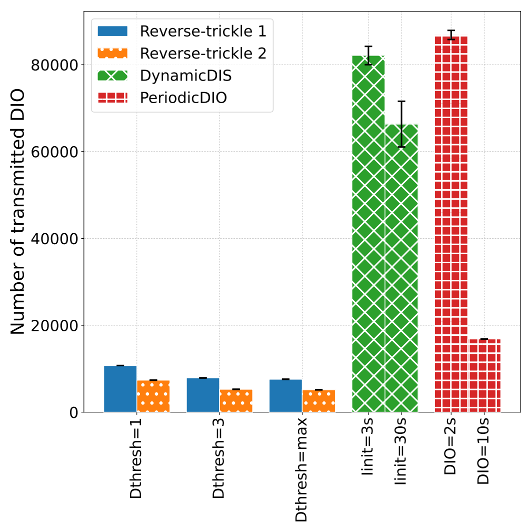



**Figure 3.10:** Signaling overhead

Control packets sent by all the nodes in the network, shown in Figure 3.10, give a clear view of how localized our solution is. Mobile nodes cause an increase of the control traffic only on nodes they connect to, not in the entire network. Using DynamicDIS, mobile nodes continuously multicasts DIS, resetting the trickle timer of neighboring nodes and thus increasing the overall control traffic until trickle converges again to a lower transmission frequency. PeriodicDIO floods the entire network with control packets, even in areas with no mobile nodes.

**Lessons learned** In light of those results, we advocate using $(I_{min1}, I_{max1})$ and $Dthresh = 3$ to balance reactivity with traffic overhead.

Nevertheless, this solution only considers network events and can not react properly to numerous changes occurring at the MAC layer. **Separating those two layers can make sense in wired networks, but such separation constrains the wireless ones.** I am convinced that MAC and network layers should work closely together to achieve the best results. I moved this idea forward in my next contribution.

### 3.2.3 MT-RPL, a cross-layer approach

**Introduction** Numerous solutions for mobility support are available at both MAC and network layers. We can bridge the gap between these two layers, leveraging information from the MAC layer to provide the routing protocol with a movement detection that does not tradeoff reactivity with control overhead. In a general fashion, a cross-layer approach that uses hints from MAC layer events should be more efficient in providing reactive movement detection. In Chapter 2, we presented X-Machiavel, a MAC protocol that grants mobile nodes better access to transmission resources. We propose to extend this protocol to trigger RPL to change the parent upon specific MAC events. This contribution is referred to as Mobility-Triggered RPL (MT-RPL) [56], [65].

**Contribution** In X-Machiavel, a mobile node can steal the channel pre-reserved by a fixed node or transmit its data to an opportunistic forwarder. MT-RPL defines a layer 2 (L2) trigger to asynchronously inform RPL every time those operations occur at the MAC layer. Upon activation, RPL initiates a parent change to reflect the effective transmission conditions: the node that received the data from the mobile node becomes its new RPL parent. To reduce the competition between multiple opportunistic forwarders together with oscillations, we limit the features of X-Machiavel (channel stealing and opportunistic forwarding) to nodes that present progress in the path towards the root. For this, MT-RPL includes the rank computed by RPL in the X-Machiavel preambles that are sent before each effective transmission.

On an idle channel, a mobile node initiates a transmission to its RPL parent by sending a preamble of type P0 (Table 2.3 details the message types used in X-Machiavel), including its RPL rank. Upon reception, the destination sends an acknowledgment of type PK1 and claims the data. If the initial destination is in range no more, another fixed node that receives the preamble may act as an opportunistic forwarder. A fixed node can acknowledge any P0 preamble that presents a rank greater than its rank with a PK0 acknowledgment. By contrast, forwarders with a rank equal to or greater than the one announced in the preamble simply discard the message. If the preamble is acknowledged by an opportunistic forwarder, the mobile node updates the MAC destination accordingly and sends the data. Those operations are illustrated in Figure 3.11.

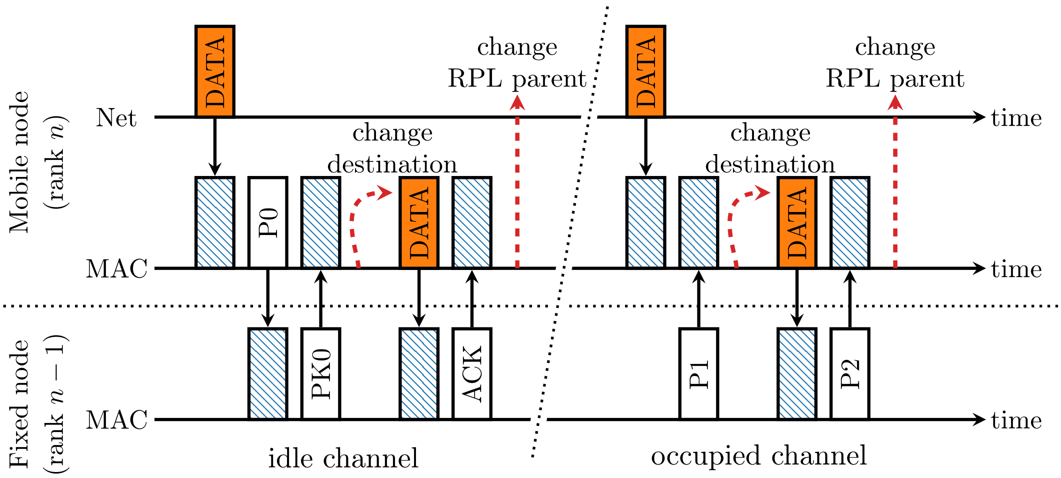



**Figure 3.11:** MT-RPL operations when the mobile node benefits from an opportunistic forwarder or steals the medium

Transmitting data on an occupied channel requires a mobile node to transmit its data between strobed preambles that are destined to another node. However, MT-RPL enables this behavior only if the rank of the sender of the preamble is lower than the rank of the mobile node. First, a mobile node should overhear a preamble of type P1, including a rank lower than its own. Then, the mobile node changes the MAC destination of its data to this sender and transmits the resulting packet between two preamble strobes. Then, the forwarder sends its data using P2 preambles to prevent the channel from being stolen again. Data from mobile nodes are also forwarded using P2 preambles. Those operations are illustrated in Figure 3.11.

MT-RPL manages the parent set regarding the information received from the MAC layer. Whenever a mobile node benefits from an opportunistic forwarder or steals the medium (and the transmission is successful), the MAC layer provides the rank and the address of the effective next hop to the RPL layer. Upon reception, RPL sets this node as the new preferred parent, computes the new rank, and sends a new DAO if required. As a result, MT-RPL should smooth network dynamics by enabling nodes to react promptly to network change without generating extra control traffic. Also, if the preamble from a mobile node is not acknowledged, MT-RPL removes the current parent and resets the rank to infinite. This turns all fixed nodes in the neighborhood into potential new parents.

**Results analysis** The evaluation of MT-RPL is done through simulation using WSNet and experimentation using IoT-LAB. We simulated a regular grid of 36 fixed nodes on a 100x100m area in which 5 mobile nodes move in this area as billiard balls [28]. Experimentations take place on the Strasbourg site of IoT-LAB. We deployed 25 fixed nodes organized as a 5x5 grid, and one mobile node roams between 2 waypoints. To increase the number of parent changes in the experimentations, we stop the service of fixed nodes toward the mobile nodes at random time intervals. The fixed nodes serve the mobile nodes between 3-5 minutes before being turned off for 1-4 minutes. This service provision allows mobile nodes to find an active preferred parent in the network.

In both the simulations and experimentations, the mobile nodes exchange data with the root at 1pkt/5s in both directions. The fixed nodes send data to the root at 1pkt/30s. We compared MT-RPL with the unreachability detection mechanisms proposed by RPL, namely Neighbor Unreachability Detection (NUD) being a part of Neighbor Discovery [53] and Bidirectional Forwarding Detection (BFD) [66]. NUD is configured with the default values. BFD exchanges probes every 30 sec., and 1 missed packet brings the session down. Simulation and experimentation parameters are further detailed in [63] and [56] respectively. We also simulated a random topology and performed experimentations on the Grenoble site of IoT-LAB but chose here to focus on the grid.

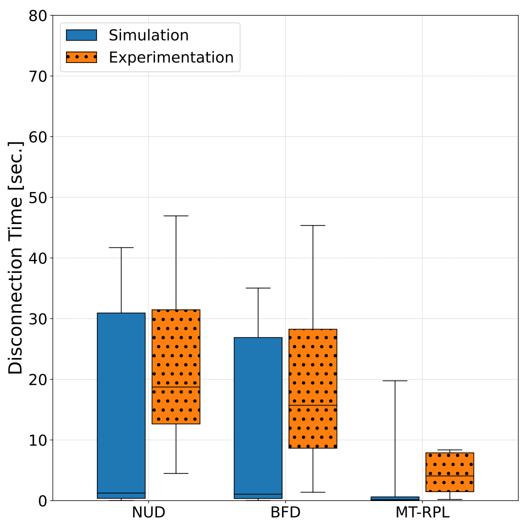



**Figure 3.12:** Disconnection time

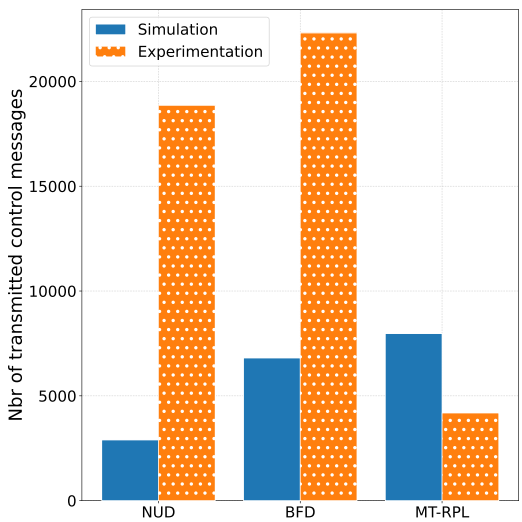



**Figure 3.13:** Signaling overhead

Figure 3.12 shows the disconnection time for each scenario. Comparatively to RPL with a stable metric, the disconnection time is drastically reduced with NUD or BFD. When the mobile node does not receive reachability confirmation from its preferred parent, RPL resets the rank to infinite and starts sending DIS. BFD lowers the maximum disconnection time thanks to its slightly lower reachable timer (30 sec. versus 38 sec.). Variation occurs because mobile nodes need to send several DIS before reconnecting to the DODAG. MT-RPL further reduces the disconnection times thanks to the interaction between the MAC and network layers. A mobile node always regains connectivity when an opportunistic node acknowledges its preamble and successfully receives the data, or when it successfully steals the medium. In those situations, the disconnection time is bound to the sending frequency of data packets and the number of preamble strobes sent before stealing the medium or opportunistic node acknowledgment.

Figure 3.13 presents the signaling overhead of each solution. Both BFD and NUD significantly increase the number of exchanged DIO. Once a mobile node is disconnected from the DODAG, it multicasts numerous DIS to solicit fresh DIO. Any fixed node that receives a multicast DIS resets its trickle timer and sends new DIO at the maximum allowed rate. This process significantly increases the number of transmitted DIO, mainly in the experimentations where multicast DIS are received by more fixed nodes than in the simulations. In addition, BFD and NUD regularly check the connectivity with the preferred parent through dedicated control messages (BFD packets or neighbor solicitations/advertisements). BFD shows the highest signaling overhead because of the number of messages required to set up a BFD session together with a lower reachability timer. By contrast, MT-RPL does not introduce new control messages, and parent changes are mainly achieved without triggering a reset of the trickle timer, thus reducing the overall signaling overhead. However, MT-RPL increases the number of unnecessary reconnections, which significantly increases the number of exchanged DAO. This is particularly visible in the simulation results.

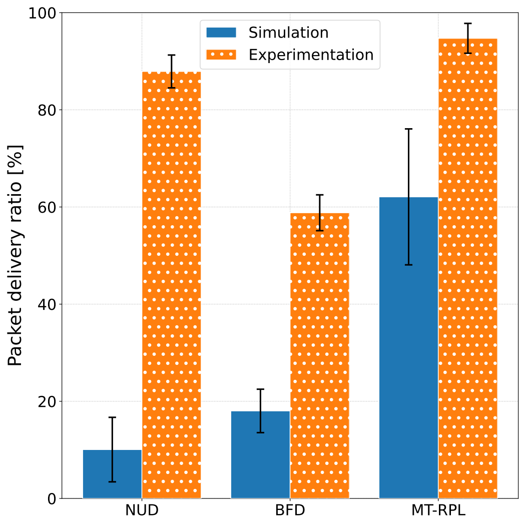



**Figure 3.14:** PDR for upward traffic

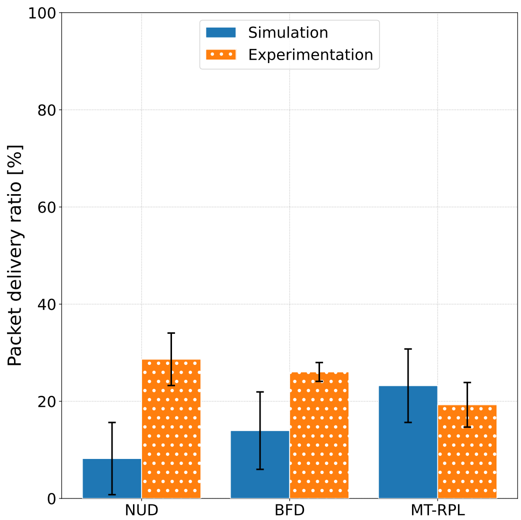



**Figure 3.15:** PDR for downward traffic

The Packet Delivery Ratio (PDR) is presented in Figure 3.14 for upward traffic (from mobile nodes to the root) and Figure 3.15 for downward traffic (from the root to mobile nodes). Lowering the disconnection time should increase the PDR of the mobile nodes. However, BFD and NUD still present low PDR due to large disconnection time (up to 47 sec.), as shown in Figure 3.12. By contrast, MT-RPL significantly increases the packet delivery ratio for upward traffic, thanks to the mechanisms favoring the medium access to mobile nodes.

However, data packets are still lost because of congestion on the path towards the root. This observation mainly concerns the simulation results.

However, the paths from the root to the mobile nodes present a low PDR for all solutions. Mobile nodes keep their preferred parents (being reachable or not) for long periods with NUD and BFD. As a result, advertised downward routes are more stable but do not necessarily reflect the current position of mobile nodes. By contrast, MT-RPL triggers numerous parent changes, with a high ratio of unnecessary ones, generating instability. An intermediate router is likely to have outdated entries or be in a *no route to host* situation.

**Lessons learned** To avoid mobile nodes from being disconnected from the graph for long periods, RPL suggests using external mechanisms such as NUD or BFD. However, we showed that neither of them allows for fast movement detection, and they significantly increase the control overhead. By contrast, MT-RPL is a cross-layer solution that reacts synchronously with MAC layer events. As a result, the mobile nodes smoothly move from one parent to another.

The results obtained for this solution confirm my intuition about cross-layer approaches and show that I am on the right track. **Exchanging information between MAC and routing layers allows a better integration of mobile nodes in LLWN.** However, this solution triggers unnecessary parent changes that make downward routes unstable. This issue remains, for now, an open challenge that I still need to address.

## 3.3 Conclusion

In this chapter, I presented the scientific challenges introduced by mobility at the network layer and investigated two approaches for mobility support. In the past few years, applications shifted from reduced mobility (e.g., healthcare appliances) to highly mobile scenarios. For example, industrial applications can involve mobile nodes moving up to 35km/h [67]. In some scenarios, the need for rapid handovers becomes of crucial importance because mobile nodes are expected to re-establish end-to-end communication within 5sec. after the connection loss [68].

To address those requirements, I first focused on using an anchor point and tunneling to relay the traffic to the current position of mobile nodes via the study of Mobile IPv6. As for broadband networks, this protocol suffers from slow movement detection. The modifications made in neighbor discovery to save energy in LLWN further increase the time required to detect the connection loss, causing serious packet loss. Several research efforts have focused on Mobile IPv6 and its variation, such as Hierarchical Mobile IPv6, Fast Mobile IPv6, or Proxy Mobile IPv6 [69]. For example, SPMIPv6 [70], [71] applies the principles of Proxy Mobile IPv6 in LLWN and reduces the disconnection time by a factor of 10, but the gains are mainly due to PMIPv6, which allows mobile nodes to keep the same address regardless of their locations. CSPMIPv6 [72] presents an enhanced architecture of SPMIPv6 that uses the concept of Hierarchical Mobile IPv6 to move the relay station closer to the mobile nodes. However, the movement detection is still based upon active solicitations from the mobile nodes, preventing them to achieve both rapid and energy efficient handovers.

Movement detection can be anticipated before the effective connection loss using link quality indicators [73], or nodes trajectory [74], [75]. However, the computation of trajectories remains complex in non-specific scenarios. In addition, the relation between a link quality indicator (such as the RSSI) and the distance remains questionable and presents stability and accuracy issues [76]. Personally, I am not in favor of pro-active solutions for asynchronous LLWN because anticipating the node movements remains a difficult task, especially on such constrained hardware, and changing its attachment point involves few signaling, which defeats the idea of *make before break*.

With the increasing popularity of RPL, the scientific community favored this protocol over dedicated solutions such as Mobile IPv6 for mobility support in LLWN [77]. However, when operated in non-storing mode, the RPL DODAG root can be seen as a relay station because all the traffic transit towards this node before being delivered to the final destination. The only difference I see here is that RPL replaces tunneling with source routing. Such an approach is in line with the current research efforts for mobility support in broadband networks for which segment routing, leveraging the source routing paradigm, is becoming an alternative to tunneling [78]. RPL relies on external mechanisms to detect the unreachability of a parent, but I showed that the proposed solutions fail to provide rapid movement detection. The solutions proposed in the literature mainly reuse the same ideas as for Mobile IPv6: they rely on link quality indicators to detect the link degradation with the current parent, such as [79]–[82], or use the positions of the mobile nodes [83], [84]. As shown with the reverse trickle algorithm, solutions based on periodic confirmation of the reachability of the parent generate extra overhead that can significantly impact energy consumption. With MT-RPL, I proposed a cross-layer approach that benefits from layer 2 events to change the parent without relying on unstable and time-varying link quality indicators, such as RSSI. Nevertheless, the disconnection time is tied up with the traffic periodicity of mobile nodes. If the period between two consecutive messages is long, the mobile nodes may be disconnected from the graph for long periods, causing packet loss in the potential downward traffic. In addition, recent studies showed that downward routes built by RPL are unreliable [16]. As MT-RPL triggers unnecessary parent changes, it may generate a storm of DAO that further degrades the reliability of downward routes.

## 3.4 Research directions

In this chapter, I investigated relay-based mobility management solutions. However, I think that IP-tunneling, as used in Mobile IPv6, is not flexible enough to transport the traffic destinated to mobile nodes. In fact, the only purpose of tunneling is to allow mobile nodes to have a stable IP address, a feature that can be guaranteed by involving the routing protocol in the process. In addition, distributed approaches [85] that scale better than Mobile IPv6 extensively use tunnels to forward traffic from old locations to new ones. Without optimizations, the triangular routing problem will be exacerbated in tree-based LLWN as implementing p2p routes remains an open challenge. For all those reasons, I advocate relying on the routing protocol because we can directly take action on the forwarding process. In addition, the presence of many routers helps to update finely routing paths.

My latest contribution supports this reasoning and reduces the disconnection time below the 5 seconds limit in most cases but may generate a storm of DAO due to numerous unnecessary parent changes. A quick fix consists of introducing a threshold for the number of transmissions that are carried out by an opportunistic forwarder from which a mobile node triggers a parent change. This threshold should be adjusted regarding the traffic periodicity. For the stability of downward routes, I am convinced that we need to deploy fast and transient mechanisms while the attachment point of a mobile node is still oscillating. For example, we can multicast packets destined to mobile nodes, as considered in [16]. Branches of the multicast tree can be added or pruned regarding the current position of the mobile node. Alternatively, parents of mobile nodes can reroute packets in case of unsuccessful transmissions, as proposed in [86].

Still, my main interest lies in segment routing [87] and how it opens the way toward complete control over the forwarding paths. In a nutshell, segment routing allows a node to enforce a list of instructions (called *segments*) to apply to a packet. An instruction is a function executed by a specific node in the network, ranging from simply forwarding the packet to a particular node (thus enabling source routing) to any complex user-defined behavior. Software Defined Networking (SDN) can leverage segment routing to encode and distribute specific instructions/functions through the network. As discussed in Chapter 5, I am convinced that SDN is the missing piece to unleash the full potential of LLWN, and computer networks in general. From a mobility support perspective, segment routing brings crucial features such as fast rerouting, buffering, bi or multicasting, and probing that can be deployed quickly to manage transient states while the routing protocol converges. I think the development of a receiver-initiated MAC protocol can be the missing piece toward the ultimate network-based solution for mobility management. Periodic beacons from mobile nodes help to track their positions, triggering updates of the routing maps whenever necessary. Segment routing also brings a large variety of actions to control such updates. As a result, the network (i.e. the fixed nodes) performs all mobility-related operations on behalf of the mobile nodes. In this context, wake-up radios can move the performance at the MAC layer even further, as studied in the next Chapter.

# Chapter 4

## Wake-up radios

This chapter presents my first contributions to wake-up radios. My previous work made me realize that we were close to the performance limits of asynchronous MAC protocols, and the only way of improving is to change the communication paradigm. In this context, wake-up radios have the potential to reshuffle the pecking order of MAC protocol for years to come. As such, I do not yet have contributions to mobility support leveraging this technology but I draw some perspectives at the end of this chapter. The work presented here is part of one Ph.D. thesis and the ANR Wake-UP project.

**Supervision**

- Sébastian Lucas Sampayo, *Polymorphic network protocol suite in heterogeneous wake-up IoT networks*, co-supervised (50%) with Thomas Noël (2018 - 2021)

**Publications:**

- S. L. Sampayo, J. Montavont and T. Noël. *LoBaPS: load balancing parent selection for RPL using wake-up radios*, in proc. of the IEEE Symposium on Computers and Communications (ISCC), 2019
- S. L. Sampayo, J. Montavont and T. Noël. *eLoBaPS: Towards Energy Load Balancing with Wake-Up Radios for IoT*, in proc. of the International Conference on Ad Hoc Networks and Wireless (AdHoc-Now), 2019
- S. L. Sampayo, J. Montavont and T. Noël. *A Performance Study of the Behavior of the Wake-Up Radio in Real-World Noisy Environments*, in proc. of the 1st Workshop on Wake-Up radio technologies for next generation wireless communications (AWAKE), in conjunction with EWSN, 2020
- S. L. Sampayo, J. Montavont and T. Noël. *REFLOOD: Reactive Routing Protocol for Wake-Up Radio in IoT*, in Elsevier Ad Hoc Networks, vol. 121, 2021

- N. El Hoda Djidi, S. L. Sampayo, J. Montavont, A. Courtay, M. Gautier, O. Berder and T. Noël, *The revenge of asynchronous protocols: Wake-up Radio-based Multi-hop Multi-channel MAC protocol for WSN*, in proc. of the IEEE Wireless Communications and Networking Conference (WCNC), 2022

## 4.1 Research context

### 4.1.1 Introduction

To reduce the energy consumption of LLWN, the radio activity is duty-cycled, alternating short active periods in which nodes can communicate with longer sleeping periods. In this thesis, we focus on asynchronous protocols that we thought are more suited to cope with network dynamics introduced by mobility. In asynchronous MAC protocols, the nodes have to periodically check the medium to detect any pending transmission. Such behavior may introduce idle listening and overhearing phenomena that can significantly impact the energy consumption. The protocol overhead (e.g., the preamble strobes used in X-MAC) also contributes to energy depletion. Finally, the main drawbacks of duty-cycling the radio is the tradeoff between latency and energy efficiency: increasing the duration of the sleeping period reduces the energy consumption but increases the latency (nodes have less transmission opportunities), and vice versa. Wake-up radios promises the end of this tradeoff.

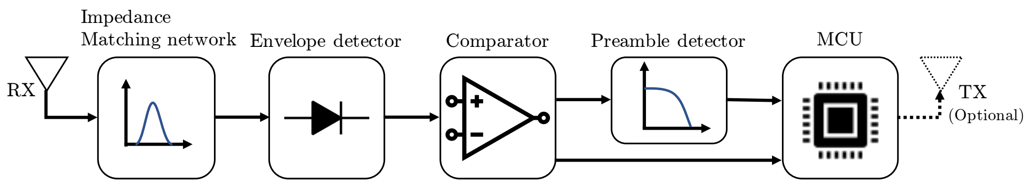



**Figure 4.1:** Block diagram of a wake-up receiver, extracted from [88]

Wake-Up Radio (WUR) is a new technology for wireless communications in LLWN. Its main characteristic is the ultra-low power consumption in listening mode, achieving between 4 and 5 orders of magnitude less than that of other technologies [89]. This is typically achieved using a simple modulation technique, such as On-Off Keying (OOK), and low datarates, in the order of 1 kbps or 10 kbps [90]. The typical block diagram of the receiver is shown in Figure 4.1. It consists of an initial impedance matching filter that maximizes the power transfer from the antenna to the circuit. Then, an envelope detector provides an output voltage proportional to the received signal energy, demodulating it into OOK. Afterward, the signal is compared to an adaptive threshold that is self-adjusted to be in the middle of the incoming signal amplitude. The resulting digital signal is then fed into an ultra-low-power microcontroller that processes the content of the signal. In parallel, the output of the comparator is used in the preamble detector to filter high-frequency noise in the channel. The output of that sub-system triggers an external interrupt of the WUR microcontroller which uses this information to validate or not the data received from the comparator.

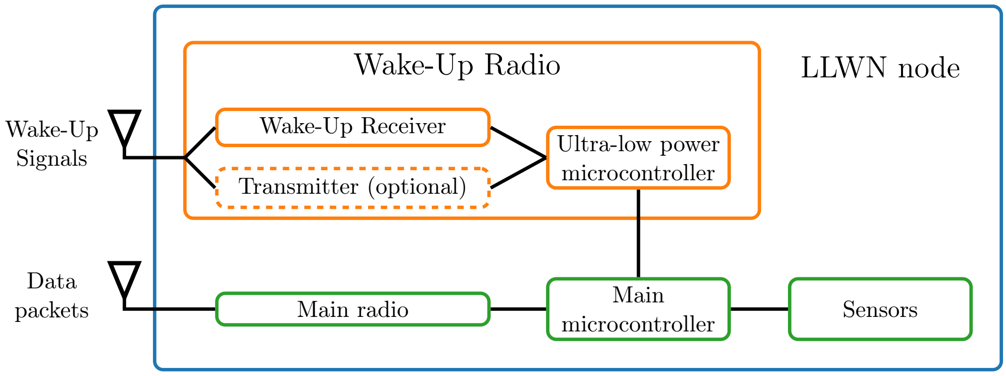



**Figure 4.2:** Architecture of an LLWN node with wake-up Radio

The ultra-low power consumption achieved by WUR comes at the cost of lower bitrate and sensitivity compared to traditional transceivers [90]. In addition, the used modulation is less robust against radio propagation effects such as interference or multipath fading. As a result, this technology is not envisioned to transmit long data packets, but complements a traditional radio transceiver to achieve energy efficiency as illustrated in Figure 4.2. Thanks to WUR, we can implement a full asynchronous MAC protocol [91]. When a node wants to communicate a data packet, it first sends a packet on the WUR channel (referred to as *Wake-Up Signal* - WUS) toward the destination. Upon reception, the destination wakes up its main radio and waits in listening mode for the data packet. After exchanging data and acknowledgment over the main radio, the destination puts back its main radio into sleep to save energy. Figure 4.3 illustrates a typical transmission using WUR. We define the term *Sync Delay* as the time between the transmission of the WUS and the effective data transmission.

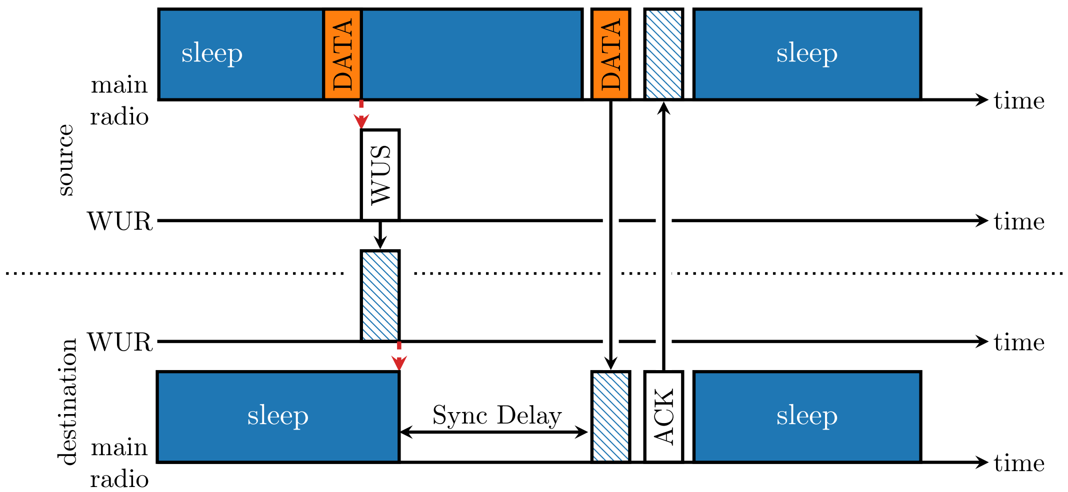



**Figure 4.3:** Example of a asynchronous transmission with wake-up radios

A wake-up signal can be either a single beam that just works as a trigger for the receiver, or modulated with digital data to convey information that is decoded by the comparator and the microcontroller submodules. Typically, such information can be the destination ID which allows receivers to filter out WUS not destined to them to avoid false wake-up.

### 4.1.2 Problem statement

One of the main scientific challenges of WUR is to tackle the range mismatch between the WUR and the main radio. The drawback of the ultra-low power consumption is that the sensitivity is dramatically decreased compared to traditional transceiver (from −110 dBm to −55 dBm). This means that if the sender transmits at the nominal power, then the range of the WUR is shorter than that of the MR and ranges from 2 to 20m [92]. This can result in situations in which two nodes can communicate together with the main radio, but not with WUR. More formally, consider the graphs denoted $G1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ where $V_1 = V_2$ but $E_1 \neq E_2$. In particular, we may have $E_1 \subset E_2$ with $G_1$ being the graph formed by WUR links and $G_2$ the one formed by main radio links. We may also state that $\forall e = (x, y) \in E_2 \mid \exists$ a WUR path with edges in $E_1$ between $x$ and $y$. Figure 4.4 illustrates $G_1$ and $G_2$.

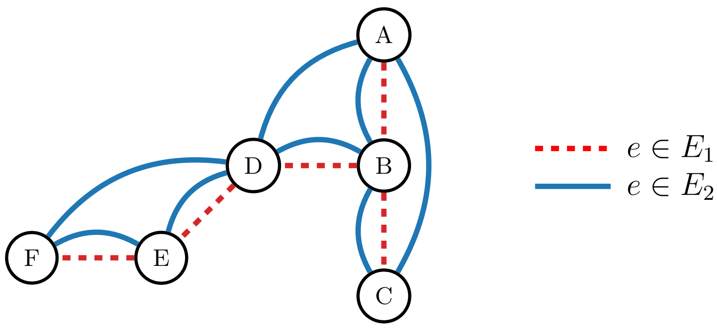



**Figure 4.4:** Graphs formed by the WUR ($G_1 = (V_1, E_1)$) and the main radio ($G_2 = (V_2, E_2)$)

Multiplying the number of hops at the main radio to only use edges in $E_1$ may defeat the gains of using WUR in the first place. Instead, we can relay WUS in $G_1$ toward the destination to minimize the number of hops in $G_2$. In the example shown in Figure 4.4, to transmit a packet to $C$, $A$ can send a WUS to $B$ that forwards it to $C$. However, the number of WUR relays between a source and a destination may have a significant impact on the network performance. In addition, relaying WUS makes difficult the computation of the Sync Delay. A longer path in $G_1$ makes the destination wake up later. So if the Sync Delay is not long enough the data will be transmitted while the destination main radio is still sleeping, resulting in a lost packet. Also, the hardware design of the WUR and the modulation used are likely to make WUR sensitive to radio propagation effects such as interference or multipath fading. This is especially challenging because of the network density and the long time over the air, due to the low datarate, that increases the contention.

To investigate these challenges, we performed simulations and experimentations using Cooja and an ad hoc deployment. In the simulation, we considered a line shaped network in which a source $S$ and a destination $D$ are in range of the main radio and we set a varying number of WUR relays between $S$ and $D$. More formally, $G_2$ is stable:

$$G_2 = (V_2, E_2) \text{ with } V_2 = \{S, D\} \text{ and } E_2 = \{(S, D)\}$$

while we define a variety of $G_1$:

- no relay: $G_1^0 = (V_1^0, E_1^0)$ with $V_1^0 = V_2$ and $E_1^0 = \{(S, D)\}$
- one relay: $G_1^1 = (V_1^1, E_1^1)$ with $V_1^1 = V_1^0 \cup \{R_1\}$ and $E_1^1 = \{(S, R_1), (R_1, D)\}$
- ...
- $n$ relays: $G_1^n = (V_1^n, E_1^n)$ with $V_1^n = V_1^{n-1} \cup \{R_n\}$ and $E_1^n = \{(S, R_1), (R_1, R_2), ..., (R_{n-1}, R_n), (R_n, D)\}$

For this analysis we fixed $n = 8$ and evaluated 3 MAC protocols. *Single WUS* reflects the scenario illustrated in Figure 4.3. We set three values for the Sync Delay: *short* (data transmission as soon as possible), *medium* and *long*. In the long case, the delay is long enough so that the WUS can hop up to 9 times (which is the maximum distance in the line-shaped network of our setup) and arrive successfully at the main destination before the main data is sent. In *WUSACK*, whenever $D$ receives a WUS, it transmits an acknowledgment (WUS ACK) back to $S$ via the WUR channel. The main data is transmitted upon reception of the WUS ACK. Finally, we include ContikiMAC [33] with the default parameters and turn on the phase optimization mechanism as a benchmark for our WUR protocols. For scenarios $G_1^1$ to $G_1^8$, the nodes $R_k$ act only as WUR relays.

During the simulation, $S$ sends 43 bytes every second to $D$. We set the probability of successful transmissions to 80% in both channels. The currents and voltage values for the main radio are the nominal values obtained from the T-Mote Sky platform [1]. The values for WUR are borrowed from the experimental validation in [90]. Simulation parameters are further detailed in [93].

We see in Figure 4.5 that the Packet Delivery Ratio (PDR) decreases with the increase of the number of WUR relays in the path toward the destination. A higher number of relays results in more WUR transmissions, decreasing the probability of successful wake up of the destination due to collisions or interference. This is clearer in the WUR + WUS ACKs scenario which present a PDR lower than 80% from 2 WUR relays. This scenario doubles the amount of packet exchanged on the WUR channel, reducing significantly the reliability of the communication. In consequence, the PDR is tightly controlled by the RX success probability

[1] https://insense.cs.st-andrews.ac.uk/files/2013/04/tmote-sky-datasheet.pdf (accessed September 26, 2022)

of the WUR medium. Therefore, we need to secure the wake up of the destination with acknowledgments, but on the main radio.

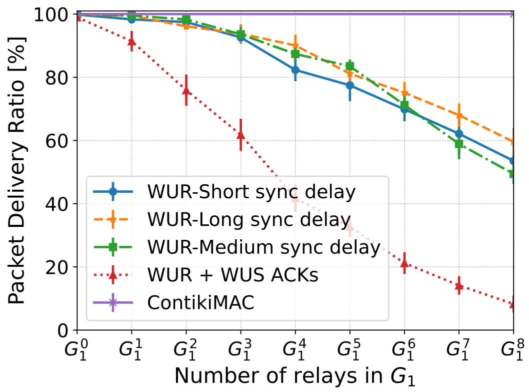



**Figure 4.5:** Packet Delivery Ratio

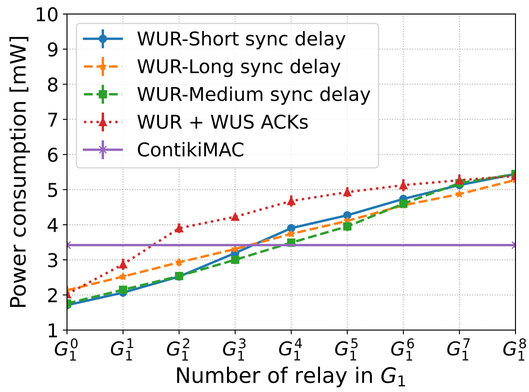



**Figure 4.6:** Network power consumption

In Figure 4.6 we compare the total power consumption of the network. Although the WUR protocols allow to drastically reduce the power consumption of each node individually, those protocols increase the network density and therefore the number of network nodes. With a number of WUR relays lower than 4 (i.e., from $G_1^0$ to $G_1^3$), the WUR schemes consume less power than ContikiMAC if we exclude the WUR + WUS ACK scenario. Having more nodes in the network is compensated by the energy efficiency of the WUR receivers. However, this trend is reversed from 4 relays. This means that having 2 nodes communicating with ContikiMAC is better in terms of overall power consumption than having 2 nodes communicating with a WUR protocol with 4 (or more) WUR relays. We can note that with 3 relays ($G_1^3$), the PDR remains above 90%.

To investigate further WUR, we performed experimentations on a network composed of three nodes: one root/gateway and two transmitters using the WUR prototype presented in [94]. A 2-bytes length signal is considered as WUS when the preamble is correctly received. Transmitters are sending a WUS every second at 1 kbps. The RX reception ratio on WUR is crucial to achieve high PDR, so we decided to provide transmitter 2 with a Clear Channel Assessment (CCA) mechanism. The CCA function takes advantage of the preamble detector module in the WUR circuit to sense activity in the channel. If a signal strong enough is received, this module generates an interrupt and turns on a busy-channel flag which postpone the local transmission.

Figure 4.7 shows the PDR achieved by each transmitter together with the number of *Negatives* and *Invalids* received per second. *Negatives* are WUS that were received but their contents differ from the original content while *Invalids* are signals with invalid preambles. We can see that there is a high correlation between the dynamic of the noise and the behavior of the PDR for both transmitters. We see that both transmitters achieve their maximum PDR when the noise is low (95% for transmitter 1 and 97.5% for transmitter 2 around 7h00).

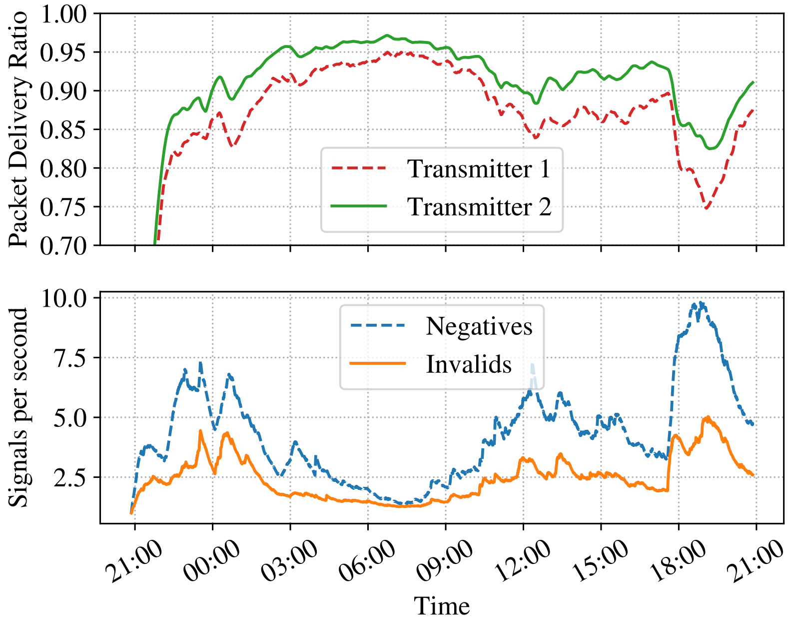



**Figure 4.7:** Packet delivery ratio and interference

Transmitter 2, with CCA, has better performance, though. This can be explained if we notice that both transmitters synchronize their transmission phase every approximately 15 mins because of the clock drift. During each synchronization period, the PDR of transmitter 1 experiences severe drops reaching down to 25% while transmitter 2 maintains a PDR higher than 85% thanks to our CCA algorithm. Moreover, transmitter 1 is more sensitive to noise peaks. When the noise level is higher, for example around 18h00, we see that the PDR of transmitter 1 decreases to 75%, while transmitter 2 maintains a PDR higher than 82%. This means an improvement of 10% when using CCA in an environment of $\lambda_{negatives} = 10$ and $\lambda_{invalids} = 5$ where $\lambda_{negatives}$ and $\lambda_{invalids}$ are respectively the mean rate of received *Negatives* per second and the mean rate of received *Invalids* per second.

### 4.1.3 Challenges

WUR networks provide a secondary communication channel but the offered datarate is too low to exchange data packets over WUR, limiting its usage to control signaling. At the MAC layer, WUR enables pure asynchronous communications in LLWN that outperforms classical asynchronous MAC protocol such as ContikiMAC in terms of latency and power consumption. Due to the range mismatch between the main radio and the WUR, we need to use relays to achieve the best performances. However, the number of WUR relays between a source and a destination have an significant impact on PDR, delays and network energy consumption. Introducing relays also requires sources to compute the *best* paths toward their destinations. If a routing structure exists, the control channel offered by WUR would bring cross-layer optimizations to help in the final routing decisions. In addition, the overall performances are very affected by the reception ratio at the WUR channel. As we experienced, transmissions over the WUR channel are prone to errors due to the usage of a simple modulation, and the low datarate increases the probability of interference. As a result, the WUS size should be reduced to the absolute minimum.

**Contribution**

To tackle those challenges, we propose to:

- define a WUR-based MAC protocol that limit the number of WUR relays to 3 and secure the WUS by acknowledging the destination wake-up over the main radio. This contribution is presented in Section 4.2.1;
- define cross-layer optimizations to help in final routing decisions. We combine the best of both worlds: the power efficiency and always-on feature of WUR with the stability of OF0 and MinHop in RPL. This contribution is presented in Section 4.2.2;
- define REFLOOD, a reactive routing protocol for clique networks that is more suited to the WUR characteristics and maximizes WUS reception ratio by using multiple paths. This contribution is presented in Section 4.2.3.

## 4.2 Main contributions

### 4.2.1 Wake-up radio-based multihop multichannel MAC protocol

**Introduction** One of the main benefits of WUR is to enable pure asynchronous communication while they limit the energy consumption. However, their lower sensitivity and their lower datarate compared with traditional IoT transceivers result in a range mismatch. WUR-based protocols should therefore use WUR relays to benefit from the maximum range offered by the main radio. In addition, WUR offers a second communication channel that can be used to negotiate network parameters before an effective transmission. In this context, our first contribution is a WUR-based multihop multichannel asynchronous MAC protocol for LLWN. This protocol, named W2M hereafter, overcomes the WUR short radio range by introducing relays and mitigates collisions over the main radio thanks to multichannel communications. We also aim at determining if a WUR-based asynchronous protocol can have significant advantages over synchronous protocols and investigate the situations where WUR prevails. For this, we selected the most popular synchronized MAC protocol of the last decade, namely Time Synchronized Channel Hopping (TSCH) [14].

**Contribution** In W2M, nodes can communicate via two communication channels. The WUR channel is only used for signaling while the main channel is used to exchange control and data packets. When a source has data to send, it first sends a Wake-Up Signal (WUS) through the WUR channel and waits for a delay (Sync Delay) to turn on its main radio. This delay corresponds to the time needed for receiving the WUS by the destination, and it depends on the number of hops and the WUS duration. Upon reception (all nodes continuously listen on this channel thanks to the WUR ultra-low power consumption), the destination turns on its main radio and broadcasts a Ready To Receive (RTR) message over the main channel, that triggers the data transmission from the source. Then, the destination sends back an Acknowledgement (ACK). In case of errors (no ACK or no RTR), the source restarts the process from the WUS transmission. To limit collisions, CSMA is used with a random backoff, and the number of packet re-transmissions is limited.

Because of the low sensibility of WUR, WUS are not necessarily received by nodes directly reachable through the main radio. To deal with this range mismatch, specific nodes are introduced as relays between the source and the destination that only serve to relay WUS. However, extending the range of WUR with such techniques is limited as using more than 3 relays could degrade the protocol performance.

To increase the network capacity, W2M uses multiple channels to exchange messages over the main radio. Before transmitting a WUS, a source randomly selects the operating channel among the sixteen IEEE 802.15.4 channels at the 2.4 GHz. As a result, each WUS includes the destination address and the operating channel of the pending communication, plus the next relay address. Note that the WUR only uses a single radio channel. The principle of this MAC protocol is shown in Figure 4.8.

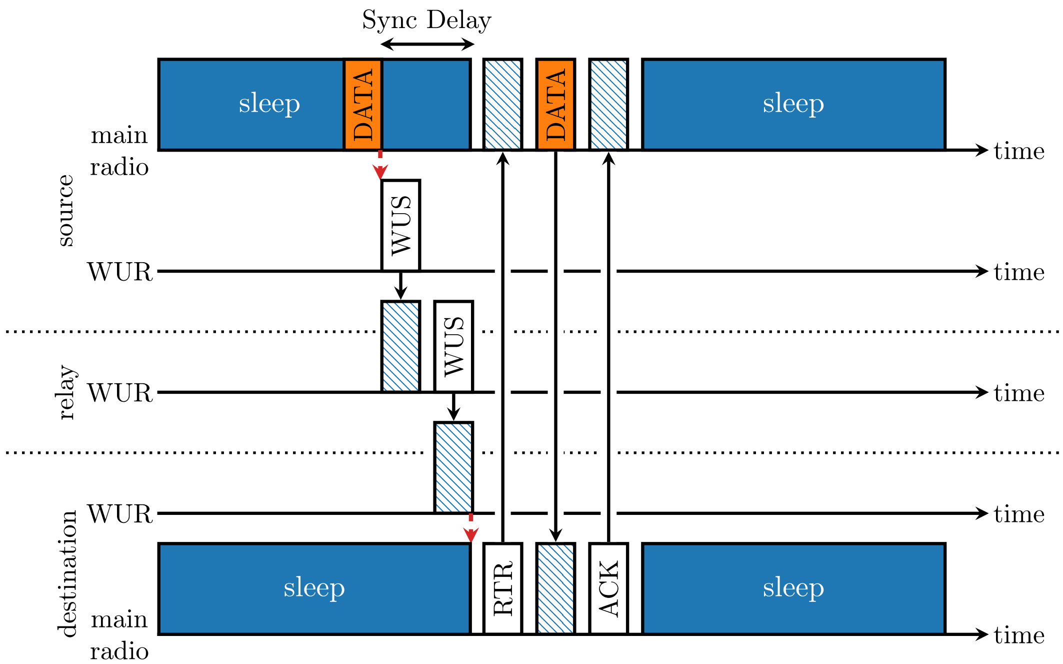



**Figure 4.8:** W2M operations

**Results analysis** We implemented W2M in Contiki using WaCo [91], a Contiki extension that provides support for WUR. We simulated a network of 30 nodes organized as a grid in Cooja. The number of neighbors of each node ranges from 2 to 4 regarding its position in the grid (only nodes located at cardinal points are neighbors). The node located at the upper left of the grid operates as the root, being the final destination of data traffic. The other nodes periodically generate a data packet of 60 bytes every 5 sec. or 2 mins, and forward it to their next hops in the grid. As we focus on the MAC layer performance, next hops are calculated offline to limit interference from a complex and high overhead routing protocol. Considering W2M, two intermediate relays are introduced between each neighbor to deal with the range mismatch between the main radio and the WUR.

We compared W2M with TSCH coupled with Orchestra [95]. Orchestra is configured with two slotframes: one of 397 timeslots (default) for the transmission of Enhanced Beacons (EB), and one of 31 timeslots for data transmission. The first is of higher priority of the latter in case of slotframe collision. The size of each slotframe allows to dedicate one timeslot to each node and therefore enables sender-based dedicated timeslots resulting in collision-free communications. Simulation parameters are further detailed in [96].

The packet delivery ratio is presented in Figure 4.9. Not a single packet is lost with TSCH in the high traffic scenario, thanks to dedicated timeslots. However, nodes located farthest away from the root lost a few packets in the low traffic scenario. On occasion, those nodes become out of synchronization and transmit while their next hops do not listen. The low traffic rate prevents periodic clock synchronization, and the drift is not always compensated by the guard time. By contrast, W2M faces more packet loss (approximately 2%) in both traffic types compared to TSCH. Collisions on WUS mainly contribute to packet loss because they are transmitted over a single channel and at a low datarate. In addition, at least 3 WUS are transmitted per communication due to relays, increasing the chance of collisions. However, once the wake-up procedure is successful, risks of collision are drastically reduced thanks to multichannel communications on the main radio.

Figure 4.10 presents the end-to-end delay for the data traffic. Delays are calculated from the generation of the data packet at the application layer of the source until it reaches the application layer of the root. Obviously, the more the number of hops between a source and the root is, the higher the delay becomes. As we can see, W2M presents lower and more stable delays than TSCH. Being purely asynchronous, W2M allows nodes to transmit at any time. The delay is only affected by the wake-up procedure (relaying WUS and broadcasting RTR) that accounts for less than a few 10 ms. Collisions explain the observed variation as a single collision increases the delay by approximately 20 ms at most. By contrast, TSCH presents larger delays (sometimes larger than 10 sec. while W2M is bounded at less than 2 sec.). In our configuration, a node can only send one data packet every 310 ms. If a node misses its timeslot, it would wait for up to 310 ms before its next transmission opportunity. This is the counterpart of collision-free communication with a static schedule as proposed

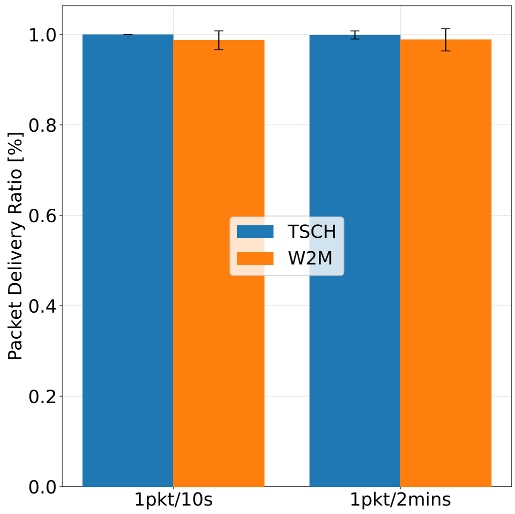



**Figure 4.9:** Packet Delivery Ratio

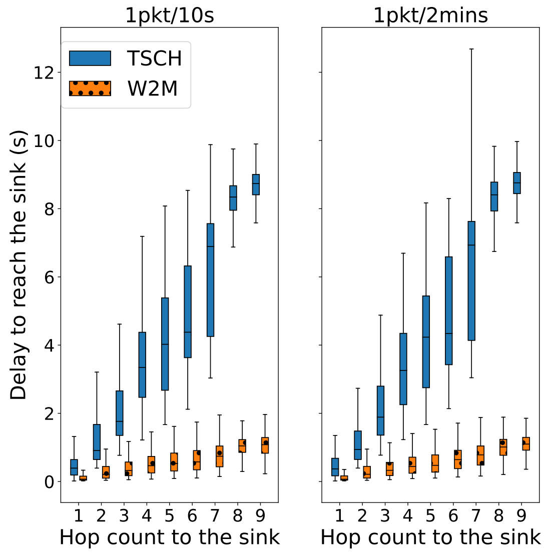



**Figure 4.10:** End-to-end delay

by Orchestra. In addition, the large variations observed for TSCH are related to limited transmission opportunities and collisions. Nodes that relay many packets are likely to queue some of them, but each queued packet takes a significant extra delay related to its rank in the queue. Finally, intra-node and inter-node collisions can still occur in TSCH. The former refers to a situation in which a collision occurs between the EB and the data slotframes, deferring the transmission of the data packet. An inter-node collision occurs when two neighbor nodes respectively transmit a data packet and an EB in the same timeslot on the same channel. Due to the collision, the data packet is scheduled for retransmission in the next slotframe, increasing the overall delay accordingly

Finally, Figure 4.11 presents the energy consumption breakdown. Nodes are grouped by the number of direct children they have in the routing path: leaf (no child), one child, and two children (the maximum). MCU is the energy consumption of the microcontroller (MCU) that performs the active tasks of the node (managing sensors, computing algorithms, acquiring data, etc.). This is typically determined by the mean current consumption of the MCU in active mode. LPM is the Low Power Mode of the main node, i.e., when the aforementioned MCU is sleeping. TX and RX are respectively the energy spent by the main radio module in transmission and reception/idle. Finally, WUR Tx is the energy spent by the WUR transmitting the WUS, WUR Rx and WUR MON are respectively the energy spent by the WUR when it is receiving a WUS and when it is only monitoring the channel.

Considering TSCH, the first factor of energy consumption is the number of direct children. As we can see, the nodes that consume the most are the ones with 2 direct children because such nodes may receive two data packets per data slotframe. Even if there is no effective transmission during those timeslots, they still switch on their radio and listen for a short

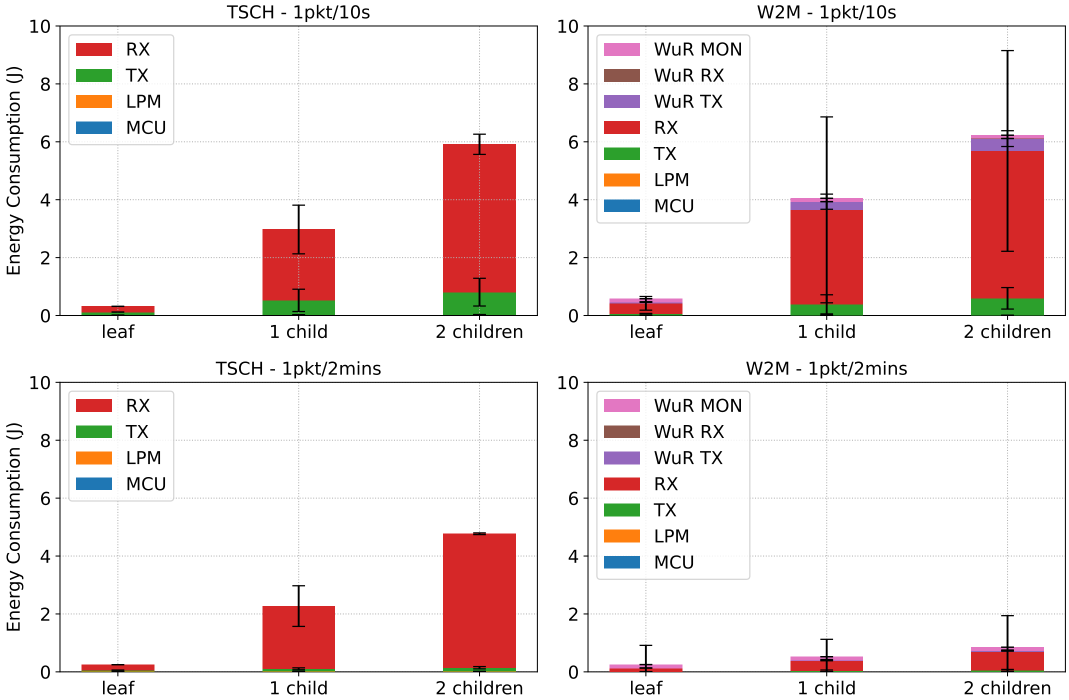



**Figure 4.11:** Energy consumption

amount of time (required to detect the packet preamble plus the guard time) before deducing that there is no packet on air. This is clearly shown in the 1pkt/2mins scenario, which presents almost the same energy consumption for RX than the 1pkt/10s scenario while 10 times less packets are exchanged. But the number of forwarded packets still has a slight impact on energy consumption. However, this factor impacts less energy consumption than the number of direct children. While the energy consumption mainly depends on the schedule in TSCH and remains almost constant in both traffic scenarios, it is directly related to the traffic in W2M. Although the consumption levels are similar between W2M and TSCH in the high traffic scenario, they are drastically reduced for W2M in the low traffic scenario (up to 78% for nodes with 2 children). In W2M, the nodes that consume the most are those that forward more packets. This explains the high variation in the energy consumption of nodes that have 1 child and 2 children because those sets include nodes with various sub-graph sizes. However, the overall energy budget of W2M could be slightly increased because of the relay nodes that are used to only relay the WUS. Nevertheless, those nodes consume between 82% and 90% less energy than nodes with 2 children thanks to the ultra-low power consumption of WUR.

**Lesson learned** W2M outperforms TSCH in terms of end-to-end delay and energy consumption at the cost of fewer guarantees on reliability. It is sweet revenge for asynchronous protocols as they are more and more abandoned in favor of synchronous protocols.

From my point of view, static TSCH schedules, such as those proposed by Orchestra, are too rigid to adapt to the traffic and can not scale without significantly increasing observed delays. I see synchronous protocols as the ultimate solution to access a wireless medium, but the best performance is only possible with an optimal schedule. **Global coordination remains essential to define such an optimal schedule but at the cost of a large signaling overhead**. By contrast, asynchronous MAC protocols allow for a distributed approach that reduces the signaling overhead and supports large-scale deployment. Thanks to WUR, their performance can be on par with synchronized protocols or go even further in some specific scenarios.

### 4.2.2 Load balancing parent selection in RPL

**Introduction** In this work, we still exploring the gains that WUR can bring to LLWN and investigated how a cross-layer solution based on WUR can add values to the network layer, RPL [10] in particular. We briefly presented this protocol in Section 3.1.2. RPL still faces several open problems: inefficient parent selection, slow recovery time after a preferred parent dies, and energy bottleneck (the preferred parent consumes way more energy than the rest of its siblings, reducing the network lifetime). The ETX metric favors reliable routes but presents serious issues with stability because of the recurrent parent changes [57]. By contrast, routes built with Hop Count are very stable but might use bad quality links [57]. Consequently, we investigated how WUR can address those issues at the MAC layer, relying on an existing routing infrastructure built by RPL. This contribution is based on the same concept as our previous contribution MT-RPL (presented in Section 3.2.3): opportunistic nodes can forward packets on behalf of the next hops computed by RPL.

**Contribution** In this work, we assume that the WUR range is the same as the main radio range. Load Balancing Parent Selection (LoBaPS) protocol takes advantage of the WUR to select opportunistic parents in RPL. LoBaPS starts operating once RPL has converged and only supports convergecast data traffic. When a source initiates a transmission, it sends first a WUS which contains its RPL rank ($R_s$) with an application ID (unique per packet to reduce duplicated packets). Upon reception, a neighbor compare $R_s$ with its own rank ($R_r$) and wakes up its main radio whenever $R_r < R_s$. This way, packets can only progress toward the root, thus avoiding routing loops.

A short time after transmitting the initial WUS, the source sends the data packet over the main channel, turns off its main radio, and starts a timer to wait for the acknowledgment over the WUR channel. When the root receives a data packet, it sends back an acknowledgment via the WUR channel. In the case of an intermediate node, it tries to forward it upwards by transmitting a new WUS with its rank. To limit collision between multiple forwarders, the MAC layer of each forwarding node computes a backoff period. The forwarder for which this

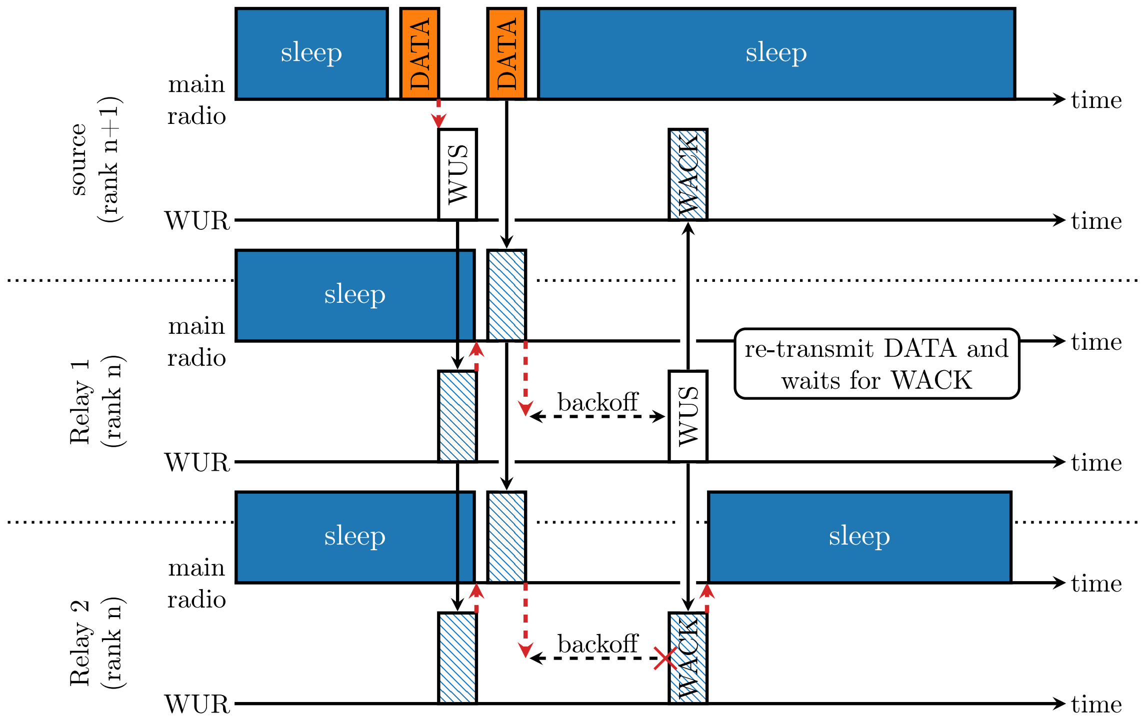



**Figure 4.12:** LoBaPS operations with two relays

period expires first can send a new WUS, including its rank. The purpose of this WUS is threefold: to wake up the next hops toward the root, to acknowledge data reception for the sender, and to cancel the backoff of the other forwarders. As a result, an acknowledgment (WACK) only differs from the WUS that triggered its transmission by the fact that the advertised rank is lower than the one included in the WUS. Those operations are illustrated in Figure 4.12 in which two relays compete to forward the data. When Relay 2 receives the WUS (acting here as WACK) from Relay 1, it cancels its backoff and drops the data.

In LoBaPS, the backoff period is randomly drawn, making the load balancing sub-optimal considering energy efficiency. In addition, there is a significant amount of energy wasted in listening mode when all the feasible successors wake up their main radio. We extended this work with eLoBaPS in which the backoff period is proportional to the energy consumed by the node so that nodes with more remaining energy have more chances to win the competition. But if the backoff is directly proportional to the amount of consumed battery, the backoff period will increase over time as the battery discharges. To keep it stable, we came up with the following equation:

$$B_j(t) = K[e_j(t) - e_{dj}(t)] + C \tag{4.1}$$

where $K$ is a constant parameter to adjust the units to milliseconds, $C$ is a small random contention window, $e_j(t)$ is the energy percentage consumed by the node $j$ at time $t$, and $e_{dj}(t)$ is the desired energy consumed at time $t$. This variable is such that if all the relays consume this energy, the load is balanced, maximizing the network lifetime. To estimate it, the nodes include their energy consumption in all the packets sent on the WUR channel (WUS and WACK). As a result, they overhear the current energy of all their neighbors. Then the value is estimated for node $j$ with a metric $r$ as:

$$e_{dj}(t) = \min_{i \in R}\{e_i(t)\} \tag{4.2}$$

where $R$ is the set of all nodes $i$ with a metric of $r$.

eLoBaPS also reduces the listening mode energy: if the current energy consumed by the node is above a certain threshold on top of the desired energy consumption, it discards the next WUS. However, this feature can create problems when exactly one parent serves one or multiple nodes. In such a situation, the packet will be delayed until the parent saves enough energy to keep up with $e_d(t)$. We set this threshold to the energy consumed by the node that has consumed the most among all its neighbors in $R$, that is:

$$e_{threshj}(t) = \max_{i \in R}\{e_i(t)\} \tag{4.3}$$

**Results analysis** We implemented LoBaPS and eLoBaPS in Contiki using WaCo [91], a Contiki extension that provides support for WUR. We compared our contributions to a reference WUR-based MAC protocol [91] presented in Section 4.1.1 and RPL. This protocol suite is referred to as W-MAC in the following. We simulated a network of 15 nodes organized as a triangular grid in Cooja. The nodes are at a maximum of 2 hops away from the root. The node density is such that each leaf can have between 2 and 7 feasible parents. The nodes send data packets of 80 bytes at different Inter Packet Intervals (IPI) set to 1, 5, 10, and 60 sec. The current and voltage values for the main radio are the nominal values obtained from the T-Mote Sky platform [2]. The values for WUR are extracted from the experimental validation in [90].

In order to simulate the battery lifetime, each node keeps track of the energy consumed, and when it reaches some maximum level, the node is shut down. This value is defined so that the root receives around 1000 packets when the first node dies. Upon node battery depletion, some nodes might become unreachable. If the graph remains connected, those nodes will reattach to the network after a certain period (e.g., the time required by the routing protocol to converge again). However, a dead node might irreversibly separate the nodes into two or more isolated subgraphs. The simulation finishes immediately and exclusively when this

[2] https://insense.cs.st-andrews.ac.uk/files/2013/04/tmote-sky-datasheet.pdf (accessed September 26, 2022)

second scenario is found. This way, it is possible to analyze the behavior of the network after the first node dies and throughout the process of parent changes. Simulation parameters are further detailed in [97] and [98].

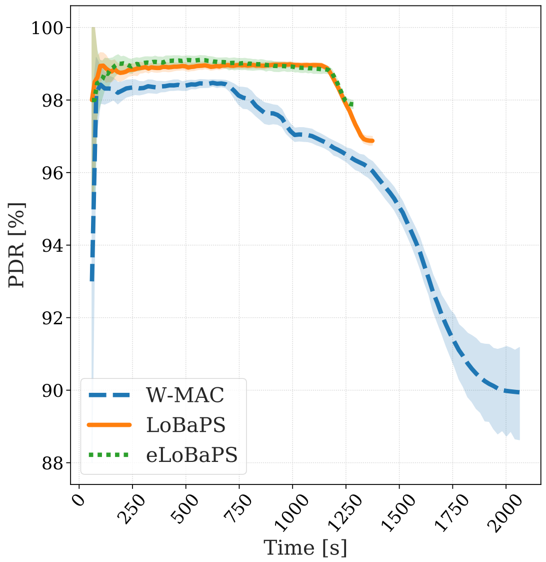



**Figure 4.13:** Packet Delivery Ratio for IPI = 10sec.

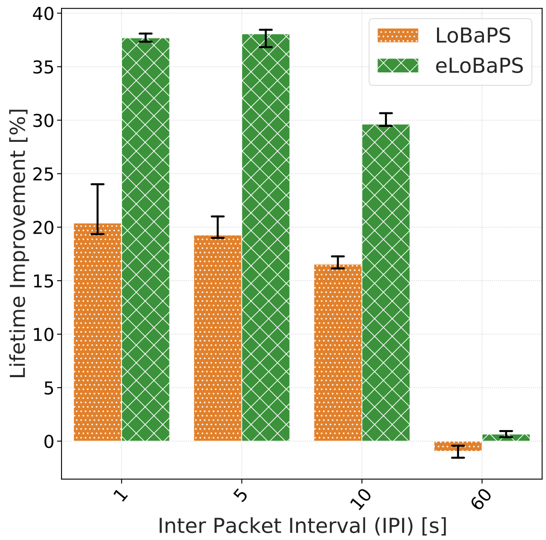



**Figure 4.14:** Lifetime improvement over W-MAC

Figure 4.13 presents the evolution of the Packet Delivery Ratio (PDR) over time for 10 sec. IPI. We can see the decline of the PDR after the first node dies, which happens between 750 and 1200 sec. for W-MAC, and around 1200 sec. for LoBaPS and eLoBaPS. In W-MAC, the PDR decreases fast and with high variability after this point, while LoBaPS presents good stability during the network lifetime and a precise and controlled decline slope. eLoBaPS improves this point by reducing the length of the decline, so there is a better ratio of stable time over decline time.

The network lifetime under different traffic loads with respect to the lifetime of W-MAC is presented in Figure 4.14. We show the relative value of the lifetime so that the results can be appreciated on the same scale for all protocols and Inter Packet Intervals (IPI). eLoBaPS outperforms LoBaPS (up to 17%) and W-MAC (up to 38%) in all scenarios. In eLoBaPS, the energy consumption is equally distributed among all the feasible parents, and nodes that are excessively consuming energy turn off their main radio to keep up with the energy consumption of their neighbors. By contrast, there are nodes that have only consumed half of their batteries when the first node with the same rank dies in W-MAC. That amount of remaining battery is the reason why the lifetime is shorter because the load is not equally shared. At the same time, almost all the protocols present very small differences for IPI set to 60 sec. The reason behind this is that as the IPI increases, the impact of radio communications on overall energy consumption decreases.

**Lessons learned** With LoBaPS and eLoBaPS we investigated how WUR can impact the performance of a whole network. For this, we used an existing routing structure, built by RPL, and showed that our solution achieves good performance. Our contribution can be extended to any routing structure that presents a notion similar to RPL ranks. The hysteresis, based on the energy in eLoBaPS, can be applied with different metrics to optimize various network parameters, for instance, the packet queue size.

WUR, thanks to the introduction of a second communication channel, can change the way we design network protocols for LLWN. However, the assumption that WUR and the main radio have the same range is unrealistic with the current advance in WUR. **Routing the wake-up signals over multiple hops, as required by W2M, remains one of the main challenges to address.** Now convinced by the technology, I started to tackle this routing problem with my next contribution.

### 4.2.3 REFLOOD: a reactive routing protocol

**Introduction** From a routing perspective, a WUR network can be viewed as a multigraph with two sets of edges: one for the WUR links and the other for the main radio links. Calculating routing paths in this context can be considered a multimetric optimization problem. However, the use of multiple metrics is non-trivial and turns out to be NP-complete [99]. In addition, the great variability of wireless links, especially for WUR, makes the task even more difficult. In this context, we investigated the problem of routing WUS toward a unique destination $D$ in a clique network (the graph formed by the main radio links is complete). More formally, we considered the situation where $G_2$ is a clique of $n$ vertices and $\frac{n(n-1)}{2}$ edges and $G_1$ is a connected graph of the same $n$ vertices but with a different set of edges as illustrated in Figure 4.15.

In computer networks, routes can be calculated with a proactive, reactive, or hybrid approach. Proactive protocols build a routing structure before any data transmission. Then, the nodes should maintain the routing structure throughout the network lifetime. RPL [10] is one of the most popular proactive routing protocols for LLWN. By contrast, reactive protocols build routes on-demand, i.e., whenever nodes have a data packet to send. LOADng [100] is an example of this approach, where the nodes request routes by flooding the network. Hybrid solutions generally use both strategies, either alternatively or at different scales [101]. The debate between proactive and reactive solutions depends on the application constraints: the complexity of reactive protocols is lower and proves to be very effective in low-traffic scenarios [102] where WUR brings the best performance. Therefore, we proposed REFLOOD, a reactive routing protocol for WUR networks.

**Contribution** REFLOOD [103] is a WUR-based routing protocol whose name combines the words *reactive* and *flood*. The idea behind REFLOOD is to flood the network with the address of the destination via the WUR channel (see *step 1* in Figure 4.15). Eventually, the flood will reach the destination, and this one will wake up its main radio to communicate as illustrated in *step 2* in Figure 4.15. This way, we take advantage of multiple paths to reach the destination and increase the chances of waking it up correctly. As a result, the impact of network dynamics due to collisions and interference on the communication channel should be mitigated.

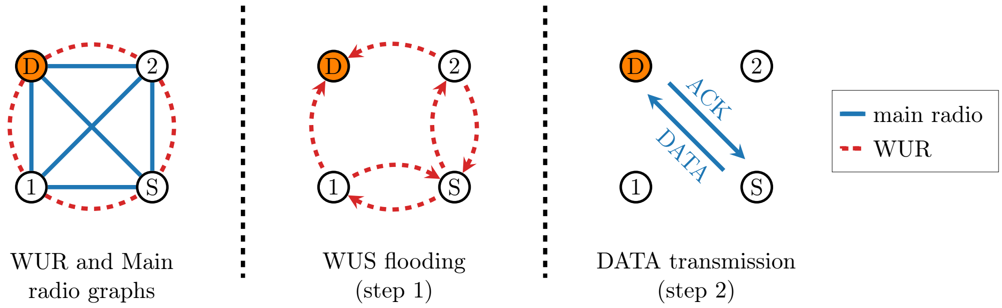



**Figure 4.15:** Data communication in REFLOOD

The classic problem of a flooding protocol is that the flood may extend to the whole network and that the nodes may retransmit the same packets infinitely. REFLOOD limits such extension by adding a hop count in the WUS. In addition, each node blocks the flooding process during a short-duration period after forwarding a WUS. This period is set to $\Delta t_{sync\_delay}$, corresponding to the SYNC_DELAY defined in Section 4.1.2. To set $\Delta t_{sync\_delay}$, we consider the maximum number of hops that the WUS can propagate ($hops_{max}$), the mean duration of the CCA backoff ($\overline{\Delta} t_{cca}$), the duration of each WUS transmission ($\Delta t_{WUS}$), and the time required to process and forward a WUS ($\Delta t_{proc}$). The mean duration of a single CCA backoff is equal to the length of the WUS in time, which means that $\overline{\Delta} t_{cca} = \Delta t_{WUS}$. The formula to compute $\Delta t_{sync\_delay}$ is given by Equation 4.4:

$$\Delta t_{sync\_delay} = hops_{max}(2\Delta t_{WUS} + \Delta t_{proc}) \tag{4.4}$$

**Result analysis** We implemented REFLOOD in Contiki using WaCo [91], a Contiki extension that provides support for WUR. We also defined PROPL to compare REFLOOD with a pro-active protocol. PROPL is similar to RPL, but the radio range of the main radio is limited to match that of WUR for all control messages. As a result, the DoDAG built and maintained by PROPL is based on the WUR topology, i.e., $G_1$. Once the protocol converged, every node has an entry in its routing table for $D$, including the WUR and the main radio next hops toward $D$. We simulated a network of 25 nodes organized as a 5x5 grid in Cooja. The unique destination $D$ is located at the top left of the grid. The other nodes act as sources and generate a packet for each period $T$ in which a uniform random variable $u$ of range $(0, T)$ is drawn to delay the effective transmission. We vary the value of $T$ to generate low ($T = 1000$ sec.), medium ($T = 60$ sec.), and high ($T = 10$ sec.) traffic scenarios. Simulation parameters are further detailed in [103].

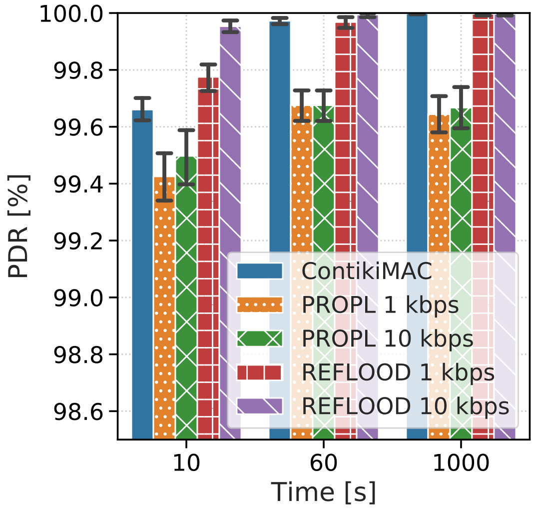



**Figure 4.16:** Packet Delivery Ratio

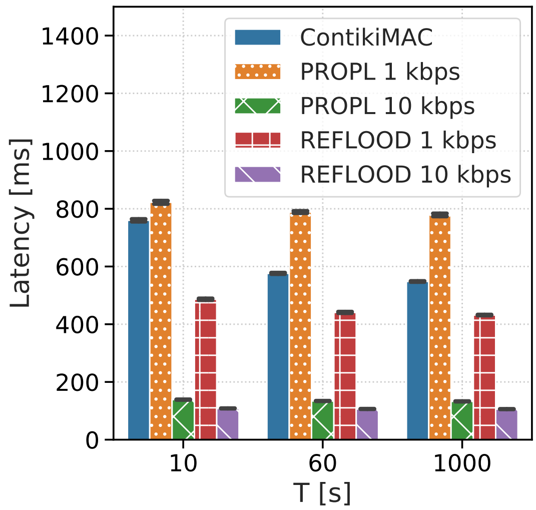



**Figure 4.17:** Latency

The Packet Delivery Ratio (PDR) is presented in Figure 4.16. REFLOOD outperforms PROPL for all traffic scenarios. In PROPL, if one WUS transmission fails on the path toward $D$ (due to a collision or interference), the whole attempt fails and triggers new retransmission. The longest WUS path includes 5 nodes, so as many possibilities to fail the transmission of the WUS. We can secure the WUS with acknowledgments, but we showed in Section 4.1.2 that using WUS ACKs is counterproductive. In REFLOOD, WUS are flooded in the network and thus are propagated through multiple paths, increasing the chance of successful delivery.

Figure 4.17 presents the end-to-end delay at the application layer. For the same datarate, REFLOOD presents lower end-to-end delays than PROPL. Reactive routing protocols generally increase the end-to-end delay because they require time to build the route before any effective transmission. Here this time is compensated by using multiple paths, while PROPL suffers from queuing effect because WUS are forwarded on a single path. Also, REFLOOD requires fewer retransmissions, which in turn results in a lower end-to-end delay.

How a protocol balances the load across the network plays an important role in the performance metrics. The goal is to reduce the maximum value of the power consumption while also distributing the load evenly among the nodes to avoid the funneling effect and extend the network lifetime. PROPL does not manage to distribute the load evenly among the nodes. The destination $D$ is overcharged because of the long idle listening on the main radio. Nodes acting as parents for many children present a higher power consumption than their siblings because they forward more WUS. Those nodes are located close to the root and around the center of the topology, typically victims of the funneling effect which contributes to the delays observed in Figure 4.17. By contrast, REFLOOD better distributes the load using

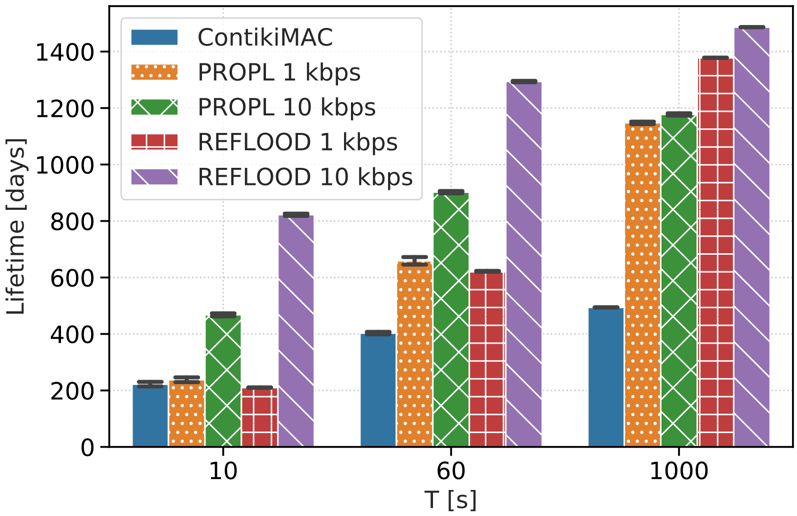



**Figure 4.18:** Network lifetime (higher is better)

multiple paths to propagate the WUS, increasing the overall network lifetime, as illustrated in Figure 4.18. The WUR datarate has a significant impact on the result because it defines the amount of time over the air required to transmit a WUS. Reducing the WUS size as much as possible is necessary because it compensates for the low datarate. In addition, the destination $D$ is also overcharged for the same reason as PROPL. This observation should be taken into consideration when extending this work to large-scale multihop networks.

**Lessons learned** The main feature is the usage of multiple paths to propagate WUS toward the destination. Such an approach improves the reliability and the end-to-end delay by eliminating the single point of failure and the funneling effect. In addition, the reactive strategy allows the reduction of the WUS size that compensates for the low WUR datarate and therefore reduces the energy consumption compared to a proactive strategy.

REFLOOD is a first step toward routing the wake-up signals over multiple hops. I think a reactive approach is a better strategy to accommodate the characteristics of WUR: links are very unstable, and the very low datarate limits the traffic capacity. In this context, **maintaining a routing structure may result in many oscillations and in increasing the energy budget far beyond the one for data forwarding.** However, I still need to scale REFLOOD to large-scale networks where nodes can be multiple hops away on the main radio (moving from a clique network to any connected graph).

## 4.3 Conclusion

The ultra-low power consumption that WUR brings to LLWN and the benefits of asynchronous MAC protocols has motivated many research groups to study the challenges of this technology [104]. At the MAC layer, the WUR can be duty-cycled and therefore uses preamble strobes similar to X-MAC to communicate [105], [106]. With recent WUR receiver [3], this can lead to 10 $\mu$A current consumption in average. However, duty-cycling the WUR will face the same challenges as duty-cycling the main radio, trading off latency for energy consumption. As observed in Section 4.1.2, relaying WUS is mandatory to benefit from the radio range of the main radio to achieve the best performance. Although electronic components continue to evolve, allowing the design of WUR prototypes with greater sensibility (e.g., $-110$ dBm [4]), the range mismatch problem between the WUR and the main radio is still open. Unfortunately, most of the proposed WUR MAC protocols focus on a single-hop network. My contribution W2M tackles this problem and competes with the most advanced synchronized protocols such as TSCH.

With LoBaPS and eLoBaPS I investigated how WUR can be used as a second control channel to provide cross-layer optimizations. OPWUM [107] and GREEN-WUP [108] are respectively very close to LoBaPS and eLoBaPS, but the former only considers a single-hop network while the latter requires to pre-compute a metric for loop avoidance. T-ROME [109] allows sources to wake up multiple relays on the path toward the destination and select the best relay, but it relies on an external tree routing protocol. To remove the need for an existing routing infrastructure, one can define a routing protocol that takes into account the two communication channels offered by WUR. CTP-WUR [110] is a proactive routing protocol that exchanges link state information across the whole network. This way, each node can compute the shortest paths toward every destination, at the main radio and WUR levels. However, collecting all this information requires exchanging numerous control messages on both channels, impacting energy consumption significantly. Other solutions [111], [112] rely on the remaining energy and hop count to select the next hops toward the destination in a convergecast scenario. However, the authors do not address the range mismatch problem, resulting in communications upper bounded by the short range of a WUR. In this context, I proposed REFLOOD, a reactive routing protocol based on WUR that tackles the range mismatch between the main radio and WUR. Thanks to forwarding the WUS on multiple paths, REFLOOD achieves a high PDR, low end-to-end delays, and significantly increases the network lifetime.

## 4.4 Research directions

The most challenging aspect of wake-up radios remains in the (very) short radio range, which limits its usage in industrial applications. In addition, current prototypes offer disparate characteristics with low reliability and are hard to get. To the best of my knowledge, only one

[3] `https://shop.radiocontrolli.com/it/wake-up-receiver/153-rc-wutrx-868.html` (accessed September 26, 2022)

[4] see footnote 3

industrial device from Controlli is available in the market. This technology still requires some innovation at the physical layer and more implications from semiconductor manufacturers to become a key enabler of IoT. Nevertheless, the WUR paradigm is making its way to other wireless technologies, such as Wi-Fi with the IEEE 802.11ba task group [113]. I am convinced that the research efforts on wake-up radio will become very competitive in the next few years. Hopefully, the work presented in this chapter can serve as building blocks for future wireless networks based on two radio-chains.

Considering W2M, I still need to improve the reliability of W2M by reducing the collisions on WUS. For example, network coding techniques [114] or backoff procedures, such as suggested in [115], can be used. Switching W2M to a receiver-initiated approach could be interesting because some studies showed that receiver-initiated protocols could perform better than WUR-based sender-initiated ones [44]. Several proposals [116], [117] already consider this paradigm in WUR. From a mobility perspective, W2M only requires slight modifications to support mobile nodes. A mobile node can solicit forwarders by sending a WUS which can be acknowledged via the *Request To Receive* (RTR), informing the source of the forwarder identity. If the channel is occupied, we can allow mobile nodes to steal the medium by adding a small delay between the RTR and the effective data transmission, similarly to X-Machiavel.

The most exciting research direction is on extending REFLOOD to support multihop networks over the main radio. My first approach is to segment the network into areas. If the network is a graph $G$, each area is a clique, i.e., an induced graph of $G$ that is complete (regarding the main radio). Areas are organized hierarchically (for example, area 0 includes the root) and interconnected via area border routers. When a node from area $N$ wants to send a packet to the root, it should push its packet to an area border router that connects areas $N$ and $N-1$. REFLOOD can be directly used for achieving this. Once the area border router receives the packet, it repeats the procedure to reach area $N-2$ and so on until reaching area 0. However, finding a maximum clique or all cliques in a given graph is one of Karp's 21 NP-complete problems [118]. Computing cliques can be achieved in exponential time [119] or in polynomial time [120] if the graph includes specific characteristics. Nevertheless, the present problem is slightly different because the clique size is constrained by the number of hops over WUR links. I envision using a hybrid approach to build and operate the network, as in ZoroMSN [101]. Areas can be built proactively with a gradient-based approach and adjacency matrix while communication in every area remains reactive using REFLOOD. Finally, areas can be provided with multiple border routers to achieve high reliability and better latency vs. energy consumption performance. However, this is likely to make the construction of areas more complex.

An alternative is to merge the two graphs $G_1$ and $G_2$ formed by each technology to a single multi-metric graph in order to compute the shortest paths toward a unique destination (we still consider a convergecast scenario). The idea is to take the graph formed by the main radio ($G_2$) and remove a set of edges for which there is no path in $G_1$ connecting the endpoints of the edge with a length lower or equal than the threshold $T_l$. More formally, let us consider the multi-metric graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ such that $\mathcal{V} = V_2 (= V_1)$, $\mathcal{E} = E_2 \setminus E'_2$ and $w : \mathcal{E} \to \mathbb{N}^2$ formed by the merge of $G_1$ and $G_2$. Note that $E'_2 = \{\{(u, v)\} \mid u, v \in \mathcal{V}$ and $u \neq v$ and $\forall p(u, v) \in G_1, len(p) > T_l\}$ such that $p(u, v)$ is a path between $u$ and $v$ and $len(p(u, v))$ is the length of the path between $u$ and $v$. In addition, $w$ is such that $\forall e = \{u, v\} \in \mathcal{E}$, we have $w(e) = (x, y)$ with $x = w_2(e)$ and $y = \min_{p(u,v) \in G_1} \{len(p(u, v))\}$ being the best WUR path between $u$ and $v$ in $G_1$. Let us assume that $G_1$ and $G_2$ are both connected and $E_1 \subseteq E_2$ ( $\forall e \in E_1 \to e \in E_2$). As a result, $\mathcal{G}$ remains connected because $E_1 \subseteq \mathcal{E}$ and we assumed that $G_1$ is connected.

As previously mentioned, computing the shortest paths using multiple metrics is non-trivial and turns out to be NP-complete [99]. I suggest first minimizing the main radio distance, i.e., considering $x$ when defined along with $y$, and secondly the WUR distance ($y$) and other objectives. For example, if there exist multiple paths $p$ between $u$ and $v$ in $\mathcal{G}$ having the same length $len(p, x)$, one can select the one minimizing the WUR distance ($len(p, y)$) and an additional arbitrary tie break. Although non necessarily optimal, I still guarantee that WUR paths present in $\mathcal{G}$ are satisfactory thanks to $T_l$. I advocate setting $T_l = 3$ regarding my findings presented in Section 4.1.2. I am currently investigating this topic with Pascal Mérindol (ICube).

I think those research directions are the necessary milestones before considering mobile nodes in WUR-based LLWN.

Chapter 5

# Conclusion and research project

## 5.1 General conclusion

Adding mobility to Low-power and Lossy Wireless Networks (LLWN) opens up new applications, ranging from Intelligent Transportation Systems (ITS) to healthcare, making our environment more and more connected. Furthermore, the number of connected objects would reach 125 billion by 2030 [2], making LLWN a key element of the future Internet. However, mobility raises serious issues at different layers of the communication stack that prevent mobile nodes to communicate correctly, resulting in disconnections, packet loss, and delays. Current communication protocols are based on concepts defined 40 years ago where mobility was impossible or not envisioned. With the emergence of wireless networking, mobility supports attracted the attention of the scientific community, and several solutions were proposed for supporting mobile nodes. This topic drives my research from the past 15 years.

I tackled this problem with a layered approach. At the MAC layer, I demonstrated that broadcasting results in synchronization issues, increasing the medium access delay together with the duty-cycle. To enforce unicast communications, I proposed to use passive overhearings to detect potential forwarders. Frequent and short listening periods achieved the best results in terms of energy consumption and successful transmission ratio but still present synchronization issues. Then, I proposed X-Machiavel which allows mobile nodes to take possession of the medium initially owned by a fixed node. This solution mitigates the losses at the MAC layer, which helps to reduce the medium access delay, the end-to-end delay, and the energy consumption. For me, we came close to the limit of asynchronous protocols, and further progress requires new communication paradigms. The (re)emergence of wake-up radios brought my attention to this technology as a means to move forward with my work on asynchronous MAC protocols. As demonstrated in Section 4.2.1, wake-up radios enables pure asynchronous protocols that challenge synchronized solutions that are currently favored by the scientific community. However, the current characteristics of wake-up radios bring new issues that should be addressed first, such as the very limited radio range of a wake-up module compared to the main radio. In this context, I proposed REFLOOD to tackle this issue for clique networks. However, REFLOOD still need to be extended to support large-scale networks before considering mobile nodes. For this, I presented two approaches that I find promising.

Transmitting data to mobile nodes involves the network layer for determining their current position. I investigated two research directions to address this problem. First, I considered relaying techniques based on namespaces that express identity and location in a single address. I also studied how the routing protocol can update the routing maps to reflect the current position of mobile nodes. In both cases, I identified the movement detection as the main cause of the disconnection time. My cross-layer approach MT-RPL (presented in Section 3.2.3) improves this process but creates unstable paths that impact the convergence of the routing protocol. My findings advocate a more flexible approach in which a fast solution should be enforced while the network is in a transient state, giving time for the routing protocol to converge. For this, segment routing opens interesting perspectives by providing complete control over the forwarding paths.

However, past and current research efforts, including some of mine, mainly reuse the abstractions, mechanisms, and protocols that popularized Internet. Today requirements are heterogeneous, ranging from best effort to strict constraints on reliability or delays. Since the 2000s, a plethora of wireless technologies appeared to satisfy or go beyond those requirements. Some of them made their way to standardization and commercial markets, while others never saw the light of day. However, each technology has its pros and cons, some focusing on the radio range, the throughput, or trading off one factor with another. LLWN emerged in the same era [1] thanks to advances in integrated circuits. Since then, numerous communication protocols have been proposed at each layer of the communication stack. For example, 47 MAC protocols are listed in [121], or more than 50 routing protocols are reported in [122]. As a result, it is very difficult to choose the protocol suite tailored for one application among this bunch of technologies and protocols. The deployment, configuration, and operation of LLWN remain complex tasks [123], especially if the optimal values are outside the range of recommended ones [124]. In this context, network providers should go through a fastidious trial and error process to fine-tune the protocol stack to meet the application needs. In addition, communication protocols are currently hardcoded in operating systems, and modifying a single parameter or switching from one protocol to another requires either recompiling and deploying new firmwares or provisioning operating systems with multiple concurrent protocols. This situation makes the adoption of new communication protocols, technologies, or services challenging. This is particularly visible in the slow-moving adoption of IPv6 which was defined more than 20 years ago and is still not supported globally.

To bring more flexibility to communication networks, LLWN in particular, I propose to shift from a monolithic network stack inherited from the TCP/IP model to programmable network functions. The emergence of Software Defined Networking (SDN) together with the wide spread of cloud computing and advances in artificial intelligence represent a great opportunity for definitely and permanently transforming how connected devices interact together. This challenging research project will drive my research efforts for the next years. I advocate an incremental approach in which we first leverage an SDN framework to optimize the LLWN performance, targeting autonomic networking. At the same time, I will focus on fully programmable network interactions in which a software engine can generate, optimize, and deploy the network blocks/functions over LLWN. Those research directions are briefly described in the next section.

## 5.2 Research project

### 5.2.1 Towards autonomic networking with software defined networking

Autonomic networking creates self-managing networks that can interface with each other and modify their behavior to provide the best service in all situations without human intervention. In recent years, Software Defined Networking (SDN) emerged as a new paradigm for making communication networks more agile [125] and forms an essential building block for autonomic networking. SDN separates the control plane (how packets should be handled) from the data plane (moving packets from sources to destinations) by removing the network intelligence from the network devices. One or multiple controllers can centralize the network intelligence to take optimal decisions on traffic forwarding. On the other hand, the network devices become simple forwarding engines, requesting the controller if new packets are received. This paradigm is particularly suited for LLWN for which the devices are constrained in terms of memory, computational power, and energy. Generally, the controller collects information about the current network status (topology, link characteristics, application requirements, etc.) and pushes actions to apply to incoming packets on network nodes via a southbound API, as illustrated in Figure 5.1. Typical actions are discarding packets, forwarding packets on specific egress interfaces, or encapsulating packets with some extra headers.

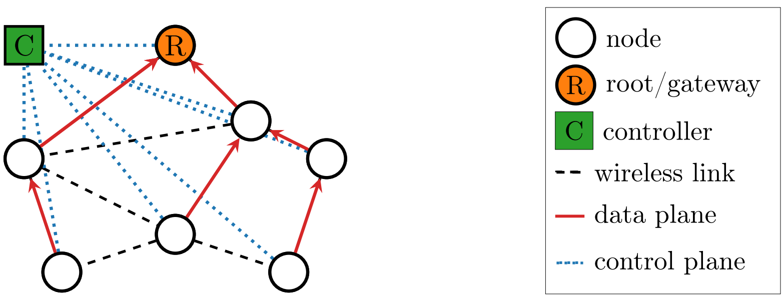



**Figure 5.1:** Example of SDN with the separation of control and data planes

In contrast to wired-based networks which benefit from point-to-point and reliable links, wireless networks manage a shared and rather a lossy medium. The work that I achieved with my colleagues over this last decade demonstrates, if proof were needed, that including the MAC layer is of crucial importance for gaining full control of LLWN. In addition, the link quality is time-variant, making discovering the topology, reporting the resource utilization, and establishing a reliable control plane challenging. In this context, I will first focus on scheduled LLWN because they already offer full control over how the wireless medium is accessed. An SDN controller can optimize the node schedule to cope with application requirements or fulfill performance metrics in terms of network capacity, bandwidth, or packet loss.

I am currently investigating this topic with Fabrice Théoleyre (ICube) in the Ph.D. thesis of Farzad Veisi. We already proposed the first blocks of an SDN solution for Time-Slotted Channel Hopping (TSCH) networks [126]. When a node connects to the network, the controller allocates new cells in the network schedule. Those cells are dedicated to the control plane to maintain a collision-free path from and to the controller. In addition, the controller performs flow admission control: new applications request the controller (via the control plane) for specific Quality of Service (QoS) requirements, such as minimum end-to-end reliability and maximum end-to-end latency. A flow is rejected if its request can not be satisfied. Otherwise, the controller provisions new cells along the forwarding path to satisfy the QoS requirements. End-to-end reliability is tackled by allocating additional cells for potential retransmissions, and end-to-end latency is minimized by allocating cells back-to-back. The configuration packet is sent over the control plane using source routing, as illustrated in Figure 5.2. Let us assume a novel flow from $A$ to $C$. Upon admission of the flow request, the controller sends a configuration packet to modify the data plane for that particular flow using source routing. Once the packet reaches $C$ (the destination of the flow), each hop enforces the cells indicated in the configuration packet in its schedule: $C$ adds cell 6, $R$ adds cells 5 and 6, $B$ adds cells 4 and 5, and finally $A$ (the source) adds cell 4. Since the configuration packet is routed from the destination to the source, the source is assured that everything has been configured when it receives the configuration packet and can safely start transmitting the data corresponding to the novel flow.

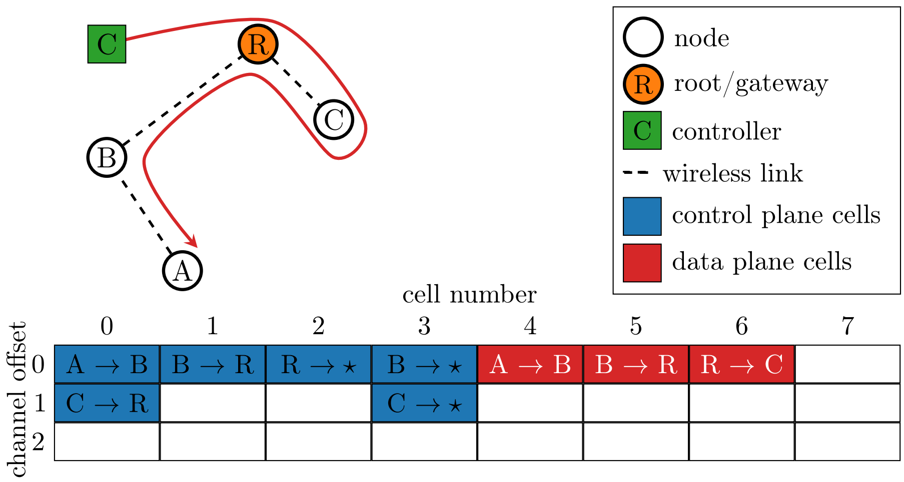



**Figure 5.2:** Data plane configuration upon flow admission from $A$ to $C$

We evaluated the performance of our SDN-TSCH solution in Cooja and highlighted how it respects per-flow guarantees, such as reliability and latency, even in the presence of best-effort traffic. This solution is an important first step, but much more needs to be done to deliver more flexible, self-adaptative LLWN. In our solution, the cells are unalterable once allocated while continuous optimization is the key to back incremental improvements of the system properties. Passive measurements are required to provide the controller with an accurate and up-to-date network view. In addition, we need efficient heuristics because computing an optimal schedule is NP-hard [127]. The reconfiguration of the data plane (or control plane) may be costly and should be initiated only if the estimated gain is considered sufficient. I envision using machine learning techniques to optimize in real-time the network schedule. In the long term, the controller could spawn simulated/emulated environments in the cloud (a digital twin) that reflects the current network status to predict the optimal network configuration regarding time-variant factors, making autonomic networking a reality.

### 5.2.2 Fully programmable networks, is this the end of protocols?

The protocol suite dedicated to LLWN networks is closely tied to the application and deployment environment, justifying the plethora of protocols for medium access control (MAC) and routing proposed in the scientific literature for more than twenty years. In this context, the arrival of SDN applied to LLWN enables a centralized network reconfiguration to continuously optimize the network performance, as detailed in the previous section. However, applications deployed over existing LLWN can evolve or shift their requirements, making the protocol suite obsolete. In addition, providers can open their networks to multiple service providers, accelerating the adoption of IoT as a Service (IoTaaS). The ability to expand in real-time the network features (e.g., network functions, communication protocols) embedded in nodes will be a key feature of future LLWN. To address this challenge, I propose to push forward the SDN paradigm with a novel device architecture enabling the programming of the whole communication stack. Gallo et al. [128], [129] already paved the way with a reprogrammable MAC. I propose going further with a complete reprogrammable stack solution (from MAC to application layers).

I am currently investigating this topic with Nicolas Montavont (IMT Atlantique), Georgios Papadopoulos (IMT Atlantique), and Thomas Noël (ICube) in the Ph.D. thesis of Amaury Bruniaux using Extended Finite State Machines (XFSM). They are Turing-complete [130] and can be used to program any communication protocol. An XFSM is a finite state machine with conditional transitions. A state represents the system status while transitions between states are actions to be executed when an event is received and the associated conditions are fulfilled. For example, Figure 5.3 presents the XFSM of the transmitter part of the *Send and Wait* protocol. When a packet is ready to send, the sender performs the transition between the *Ready* and *Wait* states, resulting in the transmission of the packet and starting a timer. If an acknowledgment is received before the expiration of the timer, the sender moves back to state *Ready*, canceling the timer. If the timer expires, the sender loops in the same state,

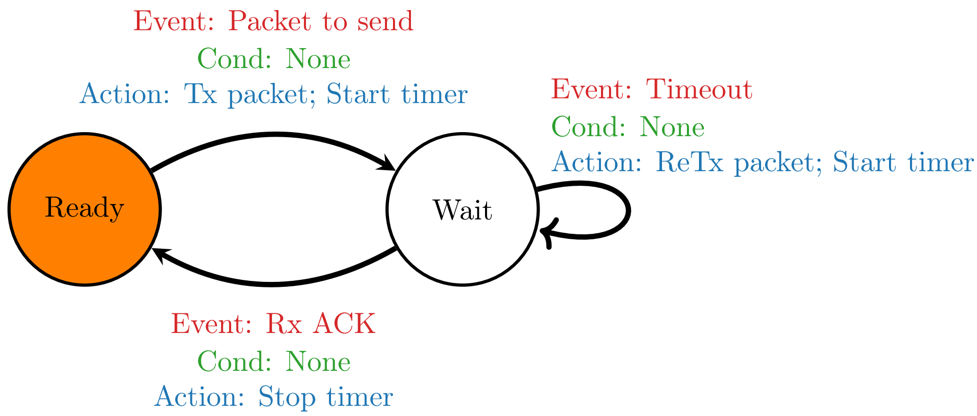



**Figure 5.3:** XFSM of the transmitter part of the send and wait protocol

retransmitting the packet and resetting the timer.

We propose a modular approach in which we use one unified XFSM that represents an entire suite of protocols. First, we decompose network tasks into a set of elementary instructions, including actions (packet management, radio control, time scheduling, forwarding, etc.), events (hardware or software signals/interrupts), conditions (boolean or arithmetic tests), and memory spaces (tables, buffers, registers). With such abstraction, we can define the whole set of communication protocols as a list of transitions that can be compiled as bytecode and executed by the nodes. As a result, network nodes become simple computing units that execute instructions similarly to typical processors. With this architecture, networks are shifting from *protocols* to *programs*, opening up infinite possibilities for more flexibility, adaptability, and scalability. For example, we can easily move from a synchronized network (e.g., TSCH) to an asynchronous one whenever necessary. Future communication protocols (MAC, routing, transport, etc.) can be easily designed and deployed on commodity hardware without complex OS-specific implementation, fostering innovation. The network can be reconfigured as a whole, or by clusters. For example, we can deploy an asynchronous protocol on the edge to smooth mobile node admission while the core is scheduled for better reliability. In addition, we plan to link the new instructions at runtime without system interruption to mitigate the impact on the network performance. Finally, this architecture also enables the deployment of network features on-demand (e.g., firewall, proxy, gateway, etc.) to adapt the behavior of specific nodes to local conditions.

This ambitious research direction is a step further toward autonomic networking: a software engine should be able to auto-generate new protocols (or should we now call them programs?) using algorithms coupled with computer-generated randomness and processing power. Machine learning techniques can estimate the program performances in simulated/emulated environments (i.e. digital twins) before real-world deployments. Advances in software-defined radio can push further this idea, making the physical layer programmable. There is a long way to go to achieve the full promise of this research project, but I am very enthusiastic to focus our research efforts on making networks fully programmable.

# References


[1] G. Pottie, “Wireless sensor networks”, in *proc. of the IEEE Information Theory Workshop (ITW)*, 1998.

[2] I. Markit, *The Internet of Things: a movement, not a market*, e-book https://cdn.ihs.com/www/pdf/IoT_ebook.pdf (accessed June 2022), 2017.

[3] J. Zheng and M. Lee, “Will IEEE 802.15.4 Make Ubiquitous Networking a Reality?: A Discussion on a Potential Low Power, Low Bit Rate Standard”, *IEEE Communication Magazine*, vol. 42, no. 6, 2004.

[4] S. Zhang and H. Zhang, “A review of wireless sensor networks and its applications”, in *proc. of the IEEE International Conference on Automation and Logistics (ICAL)*, 2012.

[5] R. Xia, C. Ye, and D. Zhang, “Vehicle to Vehicle and Roadside Sensor Communication for Intelligent Navigation”, in *Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2010.

[6] D. Popescu, C. Dragana, F. Stoican, L. Ichim, and G. Stamatescu, “A Collaborative UAV-WSN Network for Monitoring Large Areas”, *MDPI Sensors*, vol. 18, no. 12, 2018.

[7] Z. Shelby, K. Hartke, and C. Bormann, *The Constrained Application Protocol (CoAP)*, IETF Request for Comments (RFC) 7252, 2014.

[8] J. Postel, *User Datagram Protocol*, IETF Request for Comments (RFC) 768, 1980.

[9] G. Selander, J. Mattson, F. Palombini, and L. Seitz, *Object Security for Constrained RESTful Environments (OSCORE)*, IETF Request for Comments (RFC) 8613, 2019.

[10] O. Gaddour and A. Koubâa, “RPL in a nutshell: A survey”, *Elsevier Computer Networks*, vol. 56, no. 14, 2012.

[11] J. Hui and P. Thubert, *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*, IETF Request for Comments (RFC) 6282, 2011.

[12] P. Thubert, C. Bormann, L. Toutain, and R. Cragie, *IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header*, IETF Request for Comments (RFC) 8138, 2017.

[13] Q. Wang, X. Vilajosana, and T. Watteyne, *6TiSCH Operation Sublayer (6top) Protocol (6P)*, IETF Request for Comments (RFC) 8480, 2018.

[14] H. Kurunathan, R. Severino, A. Koubaa, and E. Tovar, “IEEE 802.15.4e in a Nutshell: Survey and Performance Evaluation”, *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, 2018.

[15] J. Leguay, M. Lopez-Ramos, K. Jean-Marie, and V. Conan, “An efficient service oriented architecture for heterogeneous and dynamic wireless sensor networks”, in *proc. of the IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SensApp)*, 2008.

[16] T. Istomin, O. Iova, G. Picco, and C. Kiraly, “Route or Flood? Reliable and Efficient Support for Downward Traffic in RPL”, *ACM Transactions on Sensor Networks*, vol. 16, no. 1, 2020.

[17] F. Giust, L. Cominardi, and C. Bernardos, “Distributed Mobility Management for Future 5G Networks: Overview andAnalysis of Existing Approaches”, *IEEE Communication Magazine*, vol. 53, no. 1, 2015.

[18] C. Cano, B. Bellalta, A. Sfairopoulou, and M. Oliver, “Low energy operation in WSNs: A survey of preamble sampling MAC protocols”, *Elsevier Computer Networks*, vol. 55, no. 15, 2011.

[19] P. Huang, S. Soltani, M. W. Mutla, and N. Xi, “The evolution of MAC Protocols in Wireless Sensor Networks: A Survey”, *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, 2013.

[20] J. Polastre, J. Hill, and D. Culler, “Versatile Low Power Media Access for Wireless Sensor Networks”, in *proc. of the 2nd ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2004.

[21] S. Scanzio, M. Ghazi Vakili, G. Cena, C. G. Demartini, B. Montrucchio, and A. Valenzano, “Wireless Sensor Networks and TSCH: A Compromise Between Reliability, Power Consumption, and Latency”, *IEEE Access*, vol. 8, 2020.

[22] W. Ye, J. Heidemann, and D. Estrin, “Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks”, *ACM/IEEE Transactions on Networking*, vol. 12, no. 3, 2004.

[23] A. Rajasekaran and V. Nagarajan, “Adaptive intelligent hybrid MAC protocol for wireless sensor network”, in *proc. of the IEEE International Conference on Communication and Signal Processing (ICCSP)*, 2016.

[24] Y. Liu, C. Yuen, J. Chen, and C. X., “A Scalable Hybrid MAC Protocol for Massive M2M Networks”, in *proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2013.

[25] S. Raza, T. van der Lee, G. Exarchakos, and M. Günes, “A Reliable Analysis of TSCH Protocol in a Mobile Scenario”, in *proc. of the IEEE Annual Consilper Communications & Networking Conference (CCNC)*, 2019.

[26] R. Kuntz and T. Noël, “Machiavel: Accessing the medium in mobile and dense WSN”, in *proc. of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2009.

[27] D. Roth, J. Montavont, and T. Noël, “MOBINET: Mobility Management Across Different Wireless Sensor Networks”, in *proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2011.

[28] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", *Wiley Wireless Communications and Mobile Computing (WCMC)*, vol. 2, 2002.

[29] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks", in *proc. of the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[30] R. Kuntz and T. Noël, "Machiavel: Accessing the Medium in Mobile and Dense WSN", in *proc. of the 20th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2009.

[31] R. Kuntz, J. Montavont, and T. Noël, "Improving the Medium Access in Highly Mobile Wireless Sensor Networks", *Springer Telecommunication Systems, special issue on Recent Advance in Mobile Sensor Networks*, vol. 52, no. 4, 2013.

[32] R. Kuntz, "Medium Access Control Facing the Dynamics of Wireless Sensor Networks", Ph.D. dissertation, University of Strasbourg, 2010.

[33] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol", Swedish Institute of Computer Science, Tech. Rep. T2011:13, 2011.

[34] M. Zareei, A. M. Islama, C. Vargas-Rosales, N. Mansoor, S. Goudarzi, and M. H. Rehmani, "Mobility-aware medium access control protocols for wireless sensor networks: A survey", *Elsevier Journal of Network and Computer Applications*, vol. 104, 2018.

[35] T. Zhiyong and W. Dargie, "A Mobility-Aware Medium Access Control Protocol for Wireless Sensor Networks", in *proc. of the IEEE Globecom Workshop on Heterogeneous, Multi-hop Wireless and Mobile Networks*, 2010.

[36] P. D. Ba, I. Niang, and B. Gueye, "An Optimized and Power Savings Protocol for Mobility Energy-aware in Wireless Sensor Networks", *Springer Telecommunications Systems*, vol. 55, no. 2, 2014.

[37] M. Michel and B. Quoitin, "ContikiMAC performance analysis", Computer Science Department, University of Mons, Belgium, Tech. Rep. arXiv:1404.3589v6, 2014.

[38] G. Z. Papadopoulos, A. Gallais, T. Noël, V. Kotsiou, and P. Chatzimisios, "Enhancing ContikiMAC for Bursty Traffic in Mobile Sensor Networks", in *proc. of the IEEE Sensors*, 2014.

[39] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noël, "Wireless Medium Access Control under Mobility and Bursty Traffic Assumptions in WSNs", *Springer Mobile Networks and Applications (MONET)*, 2015.

[40] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noël, "Optimizing the Handover Delay in Mobile WSNs", in *proc. of the IEEE World Forum on Internet of Things (WF-IoT)*, 2015.

[41] X. Fafoutis, A. D. Mauro, M. D. Vithanage, and N. Dragoni, "Receiver-initiated medium access control protocols for wireless sensor networks", *Elsevier Computer Networks*, vol. 76, 2015.

[42] E.-Y. A. Lin, J. Rabaey, and A. Wolisz, "Power-efficient rendez-vous schemes for dense wireless sensor networks", in *proc. of the IEEE International Conference on Communications (ICC)*, 2004.

[43] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks", in *proc. of ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2008.

[44] M. L. Wymore and D. Qiao, "RIVER-MAC: A Receiver-Initiated Asynchronously Duty-Cycled MAC Protocol for the Internet of Things", in *proc. of IEEE Annual Computer Software and Applications Conference (COMPSAC)*, 2019.

[45] J. Li, D. Zhang, L. Guo, S. Ji, and Y. Li, "ARM: an asynchronous receiver-initiated multichannel MAC protocol with duty cycling for WSNs", in *proc. of the IEEE International Performance computing and Communication Conference (IPCCC)*, 2010.

[46] D. Zuehlke, "SmartFactory—Towards a factory-of-things", *Elsevier Annual Reviews in Control*, vol. 34, 2010.

[47] M. Malik Tubaishat, P. Zhuang, Q. Qi, and Y. Shang, "Wireless sensor networks in intelligent transportation systems", *Wiley Wireless Communications & Mobile Computing*, vol. 9, no. 3, 2009.

[48] Z. Yang and C. H. Chan, "6LoWPAN Overview and Implementations", in *proc. of the International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2019.

[49] K.-S. Ki-Sik Kong, W. Lee, Y.-H. Han, M.-. Shin, and H. You, "Mobility management for all-IP mobile networks: mobile IPv6 vs. proxy mobile IPv6", *IEEE Wireless Communication*, vol. 15, no. 2, 2008.

[50] T. Camilo, P. Pinto, A. Rodrigues, J. Silva, and F. Boavida, "Mobility management in IP-based wireless sensor networks", in *proc. of the International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2008.

[51] A. Jara, R. Silva, J. Silva, M. Zamora, and A. Skarmeta, "Mobile IPv6 over Wireless Sensor Networks (6LoWPAN): Issues and Feasibility", in *proc. of the International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2010.

[52] N. Montavont and T. Noël, "Handover management for mobile nodes in IPv6 networks", *IEEE Communication Magazine*, vol. 40, no. 8, 2002.

[53] M. A. Seliem, K. M. Elsayed, and A. Khattab, "Optimized neighbor discovery for 6LoWPANs: Implementation and performance evaluation", *Elsevier Computer Communications*, vol. 112, 2017.

[54] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, *Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)*, IETF Request for Comments (RFC) 6775, 2012.

[55] D. Roth, J. Montavont, and T. Noël, "Performance Evaluation of Mobile IPv6 Over 6LoWPAN", in *proc. of the 9th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PEWASUN)*, 2012.

[56] C. Cobârzan, J. Montavont, and T. Noël, "Integrating Mobility in RPL", in *proc. of the International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2015.

[57] O. O. Iova, F. Theoleyre, and T. Noël, "Stability and efficiency of RPL under realistic conditions in Wireless Sensor Networks", in *proc. of the IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2013.

[58] J. Montavont, D. Roth, and T. Noël, "Mobile IPv6 in Internet of Things: Analysis, experimentations and optimizations", *Elsevier Ad Hoc Networks*, vol. 14, 2014.

[59] R. Silva and J. Sa Silva, *An Adaptation Model for Mobile IPv6 support in lowPANs*, IETF draft-silva-6lowpan-mipv6-00, 2009.

[60] I. El Korbi, M. Brahim, C. Adjih, and L. Saidane, "Mobility Enhanced RPL for Wireless Sensor Networks", in *proc. of the 3rd IEEE International Conference on The Network of the Future (NOF)*, 2012.

[61] T. Clausen, A. C. de Verdiere, and J. Yi, "Performance analysis of Trickle as a flooding mechanism", in *proc. of the IEEE International Conference on Communication Technology (ICCT)*, 2013.

[62] T. Winter and al., *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, IETF Request for Comments (RFC) 6555, 2012.

[63] C. Cobârzan, J. Montavont, and T. Noël, "Analysis and Performance Evaluation of RPL under Mobility", in *proc. of the IEEE Symposium on Computers and Communications (ISCC)*, 2014.

[64] K. Lee, R. Sudhaakar, L. Dai, S. Addepalli, and M. Gerla, "RPL under mobility", in *proc. of the IEEE Consumer Communications and Networking Conference (CCNC)*, 2012.

[65] C. Cobârzan, J. Montavont, and T. Noël, "MT-RPL: a cross-layer approach for mobility support in RPL", *EAI Endorsed Transactions on Internet of Things*, vol. 2, no. 5, 2016.

[66] D. Katz and D. Ward, *Bidirectional Forwarding Detection (BFD)*, IETF Request for Comments (RFC) 5880, 2010.

[67] K. Pister, P. Thubert, S. Dwars, and T. Phinney, *Industrial Routing Requirements in Low-Power and Lossy Networks*, Internet Engineering Request for Comments (RFC) 5673, 2009.

[68] J. Martocci, P. Mil, N. Riou, and W. Vermeylen, *Building Automation Routing Requirements in Low-Power and Lossy Networks*, Internet Engineering Request for Comments (RFC) 5867, 2010.

[69] M. Bouaziz and A. Rachedi, "A survey on mobility management protocols in Wireless Sensor Networks based on 6LoWPAN technology", *Elsevier Computer Communications*, vol. 74, 2016.

[70] M. M. Islam and E.-N. Huh, "Sensor Proxy Mobile IPv6 (SPMIPv6) - A Novel Scheme for Mobility Supported IP-WSNs", *MDPI Sensors*, vol. 11, no. 2, 2011.

[71] J. Kim, R. Haw, E. J. Cho, C. S. Hong, and S. Lee, "A 6LoWPAN Sensor Node Mobility Scheme Based on Proxy Mobile IPv6", *IEEE Transaction on Mobile Computing*, vol. 11, no. 12, 2012.

[72] A. Jabir, S. Shbruamaniam, Z. A. Zukarnain, and N. A. W. Abdul Hamid, "A cluster-based proxy mobile IPv6 for IP-WSNs", *EURASIP Journal on Wireless Communications and Networking*, 2012.

[73] M. HA, D. Kim, S. H. Kim, and S. Hong, "Inter-MARIO: A Fast and Seamless Mobility Protocol to Support Inter-Pan Handover in 6LoWPAN", in *proc. of the IEEE Global Telecommunications Conference (GLOBECOM)*, 2010.

[74] G. Bag, M. Taqi Raza, K.-H. Kim, and S.-W. Yoo, "LoWMob: Intra-PAN Mobility Support Schemes for 6LoWPAN", *MDPI Sensors*, vol. 9, 2009.

[75] X. Wang, Z. Dou, D. Wang, and Q. Sun, "Mobility management for 6LoWPAN WSN", *Elsevier Computer Networks*, vol. 131, 2018.

[76] L. Heurtefeux and F. Valois, "Is RSSI a good choice for localization in Wireless Sensor Network?", in *proc. of the IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2012.

[77] P. O. Kamgueu, E. Nataf, and T. D. Ndie, "Survey on RPL enhancements: A focus on topology, security and mobility", *Elsevier Computer Communications*, vol. 120, 2018.

[78] S. Matsushima, C. Filsfils, M. Kohno, P. Camarillo, D. Voyer, and C. Perkins, *Segment Routing IPv6 for Mobile User Plane*, Internet Engineering Task Force, work in progress, draft-ietf-dmm-srv6-mobile-uplane-21, 2022.

[79] F. Gara, L. Saad, E. Hamida, B. Tourancheau, and R. Ayed, "An adaptive timer for RPL to handle mobility in wireless sensor networks", in *proc. of the International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016.

[80] H. Fotouhi, D. Moreira, and M. Alves, "mRPL: boosting mobility in the Internet of Things", *Elsevier Ad Hoc Networks*, vol. 26, 2015.

[81] H. Fotouhi, D. Moreira, M. Alves, and P. Yomsi, "mRPL+: a mobility management framework in RPL/6LoWPAN", *Elsevier Computer Communications*, vol. 104, 2017.

[82] M. Bouaziz, A. Rachedi, A. Belghith, M. Berbineau, and S. Al-Ahmadi, "EMA-RPL: Energy and mobility aware routing for the Internet of Mobile Things", *Elsevier Future Generation Computer Systems*, vol. 97, 2019.

[83] O. Gaddour, A. Koubâa, and M. Abid, "Quality-of-service aware routing for static and mobile IPv6-based low-power and lossy sensor networks using RPL", *Elsevier Ad Hoc Networks*, vol. 33, 2015.

[84] M. Barcelo, A. Correa, J. Vicario, A. Morell, and X. Vilajosana, “Addressing mobility in RPL with position assisted metrics”, *IEEE Sensors Journal*, vol. 16, no. 7, 2016.

[85] M. Gohar, J. Choi, S. Koh, K. Naseer, and S. Jabbar, “Distributed Mobility Management in 6LoWPAN-Based Wireless Sensor Networks”, *Sage International Journal of Distributed Sensor Networks*, vol. 11, no. 10, 2015.

[86] S. Min, S. Chung, H. Lee, and Y. Ha, “Downward traffic retransmission mechanism for improving reliability in RPL environment supporting mobility”, *International Journal of Distributed Sensor Networks*, vol. 16, no. 1, 2020.

[87] P. Ventre, S. Salsano, M. Polverini, *et al.*, “Segment Routing: A Comprehensive Survey of Research Activities, Standardization Efforts, and Implementation Results”, *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, 2021.

[88] S. L. Sampayo, “Polymorphic network protocol suite in heterogeneous wake-up IoT networks”, Ph.D. dissertation, University of Strasbourg, 2021.

[89] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli, “Ultra low power wake-up radios: A hardware and networking survey”, *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, 2017.

[90] M. Magno, V. Jelicic, B. Srbinovski, V. Bilas, E. Popovici, and L. Benini, “Design, implementation, and performance evaluation of a flexible low-latency nanowatt wake-up radio receiver”, *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, 2016.

[91] R. Piyare, T. Istomin, and A. L. Murphy, “WaCo: A Wake-Up Radio COOJA Extension for Simulating Ultra Low Power Radios”, in *proc. of the International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2017.

[92] S. Basagni, F. Ceccarelli, C. Petrioli, N. Raman, and A. V. Sheshashayee, “Wake-up Radio Ranges: A Performance Study”, in *proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2019.

[93] S. L. Sampayo, J. Montavont, F. Prégaldiny, and T. Noël, “Is Wake-Up Radio the Ultimate Solution to the Latency-Energy Tradeoff in Multi-hop Wireless Sensor Networks?”, in *proc. of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2018.

[94] T. Polonelli, M. Magno, and L. Benini, “Poster Abstract: An Ultra-Low Power Wake up Radio with Addressing and Retransmission Capabilities for Advanced Energy Efficient MAC Protocols”, in *proc. of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2016.

[95] S. Duquennoy, B. AlNahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH”, in *proc. of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2015.

[96] N. El Hoda Djidi, S. Sampayo, J. Montavont, *et al.*, “The revenge of asynchronous protocols: Wake-up Radio-based Multi-hop Multi-channel MAC protocol for WSN”, in *proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2022.

[97] S. L. Sampayo, J. Montavont, and T. Noël, “eLoBaPS: Towards Energy Load Balancing with Wake-Up Radios for IoT”, in *proc. of the International Conference Ad-Hoc Networks and Wireless (AdHoc-Now)*, 2019.

[98] S. L. Sampayo, J. Montavont, and T. Noël, “LoBaPS: Load Balancing Parent Selection for RPL Using Wake-Up Radios”, in *proc. of the IEEE International Symposium on Computer Communications (ISCC)*, 2019.

[99] Z. Wang and J. Crowcroft, “Quality-of-service routing for supporting multimedia applications”, *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, 1996.

[100] T. Clausen, J. Yi, and U. Herberg, “Lightweight On-demand Ad hoc Distance-vector Routing - Next Generation (LOADng): Protocol, extension, and applicability”, *Elsevier Computer Networks*, vol. 126, 2017.

[101] N. Nasser, A. Al-Yatama, and K. Sale, “Zone-based routing protocol with mobility consideration for wireless sensor networks”, *Springer Telecommunication Systems*, vol. 52, 2013.

[102] J. Tripathi, J. C. de Oliveira, and J. P. Vasseur, “Proactive versus reactive routing in low power and lossy networks: Performance analysis and scalability improvements”, *Elsevier Ad Hoc Networks*, vol. 23, 2014.

[103] S. Sampayo, J. Montavont, and T. Noël, “REFLOOD: Reactive Routing Protocol for Wake-Up Radio in IoT”, *Elsevier Ad Hoc Networks*, vol. 121, 2021.

[104] M. Ghribi and A. Meddeb, “Survey and taxonomy of MAC, routing and cross layer protocols using wake-up radio”, *Elsevier Journal of Network and Computer Applications*, vol. 149, 2020.

[105] J. Junaid Ansari, X. Zhang, and P. Mahonen, “Multi-radio medium access control protocol for wireless sensor networks”, *International Journal of Sensor Networks*, vol. 8, no. 1, 2010.

[106] N. S. Mazloum and O. Edfors, “DCW-MAC: An Energy Efficient Medium Access Scheme Using Duty-Cycled Low-Power Wake-Up Receivers”, in *proc. of the IEEE Conference on Vehicular Technology (VTC)*, 2011.

[107] F. A. Aoudia, M. Gautier, and O. Berder, “OPWUM: Opportunistic MAC Protocol Leveraging Wake-Up Receivers in WSNs”, *Hindawi Journal of Sensors*, 2016.

[108] C. Petrioli, D. Spenza, P. Tommasino, and A. Trifiletti, “A Novel Wake-Up Receiver with Addressing Capability for Wireless Sensor Nodes”, in *proc. of the IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2014.

[109] T. Kumberg, M. Schink, L. Reindl, and C. Schindelhauer, “T-ROME: A simple and energy efficient tree routing protocol for low-power wake-up receivers”, *Elsevier Ad Hoc Networks*, vol. 59, 2017.

[110] A. Sheshashayee and S. Basagni, “Multi-hop Wake-up Radio Relaying for the Collection Tree Protocol”, in *proc. of the IEEE Vehicular Technology Conference (VTC)*, 2019.

[111] G. Koutsandria, V. Di Valerio, D. Spenza, S. Basagni, and C. Petrioli, “Wake-up radio-based data forwarding for green wireless networks”, *Elsevier Computer Communications*, vol. 160, 2020.

[112] S. Basagni, V. Di Valerio, G. Koutsandria, and C. Petrioli, “Wake-up Radio-enabled Routing for Green Wireless Sensor Networks”, in *proc. of the IEEE Vehicular Technology Conference (VTC)*, 2017.

[113] D. Deng, S. Lien, C. Lin, M. Gan, and H. Chen, “IEEE 802.11ba Wake-Up Radio: Performance Evaluation and Practical Designs”, *IEEE Access*, vol. 8, 2020.

[114] N. El Hoda Djidi, M. Gautier, A. Courtay, and O. Berder, “Enhancing Wake-Up Receivers Reliability through Preamble Filtering and Minimum Energy Coding”, in *proc. of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021.

[115] D. Ghose and F. Li, “Enabling backoff for SCM wake-up radio: protocol and modeling”, *IEEE Communication Letter*, vol. 21, no. 5, 2017.

[116] J. M. Lebreton, S. Kandukuri, N. Murad, and R. Lorion, “An Energy-Efficient Duty-Cycled Wake-Up Radio Protocol for Avoiding Overhearing in Wireless Sensor Networks”, *Wireless Sensor Network*, vol. 8, no. 8, 2016.

[117] T. N. Le, A. Pegatoquet, and M. Magno, “Asynchronous on demand MAC protocol using wake-up radio in wireless body area network”, in *proc. of the IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*, 2015.

[118] R. Karp and J. Thatcher, “Reducibility among combinatorial problems”, *Springer Complexity of Computer Computations*, 1972.

[119] C. Bron and J. Kerbosch, “Algorithm 457: finding all cliques of an undirected graph”, *Communications of the ACM*, vol. 16, no. 9, 1973.

[120] E. Tomita and H. Nakanishi, “Polynomial-Time Solvability of the Maximum Clique Problem”, in *proc. of the ACM International Conference on European Computing (ECC)*, 2009.

[121] A. Kumar, M. Zhao, K. Wong, Y. Guan, and P. Chong, “A Comprehensive Study of IoT and WSN MAC Protocols: Research Issues, Challenges and Opportunities”, *IEEE Access*, vol. 6, 2018.

[122] N. Pantazis, S. Nikolidakis, and D. Vergados, “Energy-Efficient Routing Protocols in Wireless Sensor Networks: A Survey”, *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, 2013.

[123] C. Zhu, C. Zheng, M. Shu, and G. Han, “A survey on coverage and connectivity issues in wireless sensor networks”, *Elsevier Journal of Network and Computer Applications*, vol. 35, no. 2, 2012.

[124] G. Anastasi, M. Conti, and M. Di Francesco, "A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks", *IEEE Transactions in Industrial Informatics*, vol. 7, no. 1, 2011.

[125] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN)", *Elsevier Computer Networks*, vol. 112, 2017.

[126] F. Veysi, J. Montavont, and F. Théoleyre, "SDN-TSCH: Enabling Software Defined Networking for Scheduled Wireless Networks with Traffic Isolation", in *proc. of the IEEE Symposium on Computers and Communications (ISCC)*, 2022.

[127] G. Sharma, R. Mazumdar, and N. Shroff, "On the complexity of scheduling in wireless networks", in *proc. of the ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2006.

[128] P. Gallo, A. Krasilov, A. Lyakhov, I. Tinnirello, and G. Bianchi, "Breaking layer 2: A new architecture for programmable wireless interfaces", in *proc. of the International Conference on ICT Convergence (ICTC)*, 2012.

[129] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli, "Wireless MAC processors: Programming MAC protocols on commodity Hardware", in *proc. of the IEEE International Conference on Computer Communications (INFOCOM)*, 2012.

[130] C. Wang and M. Liu, "A Test Suite Generation Method for Extended Finite State Machines Using Axiomatic Semantics Approach", in *proc. of the IFIP TC6/WG6.1 International Symposium on Protocol Specification, Testing and Verification*, 1992.

[131] S. Sampayo, J. Montavont, and T. Noël, "A Performance Study of the Behavior of the Wake-Up Radio in Real-World Noisy Environments", in *proc. of the Workshop on Wake-Up radio technologies for next generation wireless communications (AWAKE), in conjunction with EWSN 2020*, 2020.

[132] C. Perkins, D. Johnson, and J. Arkko, *Mobility Support in IPv6*, IETF Request for Comments (RFC) 6275, 2011.

[133] Z. Haas and M. Peralman, "The performance of query control schemes for the zone routing protocol", *IEEE/ACM Transactions on Networking*, 2001.

[134] D. Pancaroglu and S. Sen, "Load balancing for RPL-based Internet of Things: A review", *Elsevier Ad Hoc Networks*, vol. 116, 2019.

[135] C. Guo, L. Zhong, and J. Rabaey, "Low Power Distributed MAC for Ad Hoc Sensor Radio Networks", in *proc. of the IEEE Global Communications Conference (GLOBECOM)*, 2001.

[136] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC Essentials for Wireless Sensor Networks", *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, 2010.

# Appendix A

## List of Publications

### International journals


[1] S. Sampayo, J. Montavont, and T. Noël, “REFLOOD: Reactive Routing Protocol for Wake-Up Radio in IoT”, *Elsevier Ad Hoc Networks*, vol. 121, 2021.

[2] C. Cobârzan, J. Montavont, and T. Noël, “MT-RPL: a cross-layer approach for mobility support in RPL”, *EAI Endorsed Transactions on Internet of Things*, vol. 2, no. 5, 2016.

[3] J. Beaudaux, A. Gallais, J. Montavont, T. Noël, D. Roth, and E. Valentin, “Thorough Empirical Analysis of X-MAC over a Large Scale Internet of Things Testbed”, *IEEE Sensors Journal, Special Issue on Internet of Things (IoT): Architecture, Protocols and Services*, vol. 14, no. 2, 2014.

[4] J. Montavont, D. Roth, and T. Noël, “Mobile IPv6 in Internet of Things: Analysis, Experimentations and Optimizations”, *Elsevier Ad Hoc Networks*, vol. 14, 2014.

[5] R. Kuntz, J. Montavont, and T. Noël, “Improving the Medium Access in Highly Mobile Wireless Sensor Networks”, *Springer Telecommunication Systems, special issue on Recent Advance in Mobile Sensor Networks*, vol. 52, no. 4, 2013.

[6] R. Kuntz, J. Montavont, and T. Noël, “Multihoming in IPv6 Mobile Networks: Progress, Challenge and Solutions”, *IEEE Communication Magazine*, vol. 51, no. 1, 2013.

[7] R. Kuntz, J. Montavont, G. Schreiner, D. Binet, and T. Noël, “An Improved Network Mobility Service for Intelligent Transportation Systems”, *Wiley Wireless Communications and Mobile Computing*, vol. 11, no. 7, 2011.

[8] E. Ivov, J. Montavont, and T. Noël, “Thorough Empirical Analysis of the IETF FMIPv6 Protocol over IEEE 802.11 Networks”, *IEEE Wireless Communications, Special Issue on Architectures and Protocols for Mobility Management in All-IP Mobile Networks*, vol. 15, no. 2, 2008.

[9] J. Montavont, E. Ivov, T. Noël, and K. Guillouard, “Analysis of a Geolocation-based FMIPv6 Extension for Next Generation Wireless LANs”, *Ubiquitous Computing and Communication Journal*, vol. 2, no. 5, 2007.

## International conferences

[10] A. Bruniaux, J. Montavont, T. Noël, G. Papadopoulos, and N. Montavont, "Towards a fully programmable internet of things", in *proc. of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2022.

[11] N. El Hoda Djidi, S. Sampayo, J. Montavont, *et al.*, "The revenge of asynchronous protocols: Wake-up Radio-based Multi-hop Multi-channel MAC protocol for WSN", in *proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2022.

[12] F. Veisi, J. Montavont, and F. Théoleyre, "SDN-TSCH: Enabling Software Defined Networking for Scheduled Wireless Networks with Traffic Isolation", in *proc. of the IEEE Symposium on Computers and Communications (ISCC)*, 2022.

[13] B. Foubert and J. Montavont, "Sharing is Caring a Cooperation Scheme for RPL Network Resilience and Efficiency", in *proc. of the IEEE International Symposium on Computer Communications (ISCC)*, 2019.

[14] S. Sampayo, J. Montavont, and T. Noël, "eLoBaPS: Towards Energy Load Balancing with Wake-Up Radios for IoT", in *proc. of the International Conference on Ad-Hoc Networks and Wireless (AdHoc-Now)*, 2019.

[15] S. Sampayo, J. Montavont, and T. Noël, "LoBaPS: Load Balancing Parent Selection for RPL Using Wake-Up Radios", in *proc. of the IEEE International Symposium on Computer Communications (ISCC)*, 2019.

[16] S. Sampayo, J. Montavont, F. Prégaldiny, and T. Noël, "Is Wake-Up Radio the Ultimate Solution to the Latency-Energy Tradeoff in Multi-hop Wireless Sensor Networks?", in *proc. of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2018.

[17] P. Neumann, J. Montavont, and T. Noël, "Indoor Deployment of Low-Power Wide Area Networks (LPWAN): a LoRaWAN case study", in *proc. of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016.

[18] Q. Nguyen, J. Montavont, N. Montavont, and T. Noël, "RPL Border Router Redundancy in the Internet of Things", in *proc. of the International Conference on Ad-Hoc Networks and Wireless (AdHoc-Now)*, 2016.

[19] C. Cobârzan, J. Montavont, and T. Noël, "Integrating Mobility in RPL", in *proc. of the International Conference on Embedded Wireless Systems and Networks (EWSN)*, 2015.

[20] J. Montavont, C. Cobârzan, and T. Noël, "Theoretical Analysis of IPv6 Stateless Address Autoconfiguration in Low-power and Lossy Wireless Networks", in *proc. of the IEEE International Conference on Computing & Communication Technologies - Research, Innovation, and Vision for Future (RIVF)*, 2015.

[21] C. Cobârzan, J. Montavont, and T. Noël, “Analysis and Performance Evaluation of RPL under Mobility”, in *proc. of the IEEE Symposium on Computers and Communications (ISCC)*, 2014.

[22] D. Roth, J. Montavont, and T. Noël, “Performance Evaluation of Mobile IPv6 Over 6LoWPAN”, in *proc. of the ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PEWASUN)*, 2012.

[23] J. Beaudaux, A. Gallais, J. Montavont, and T. Noël, “LIFT: Layer Independent Fault Tolerance Mechanism for Wireless Sensor Networks”, in *proc. of the IEEE Vehicular Technology Conference (VTC Spring)*, 2011.

[24] D. Roth, J. Montavont, and T. Noël, “MOBINET: Mobility Management Across Different Wireless Sensor Networks”, in *proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2011.

[25] D. Roth, J. Montavont, and T. Noël, “Overhearing for Congestion Avoidance in Wireless Sensor Networks”, in *proc. of the International Conference on Ad Hoc and Wireless (AdHoc-Now)*, 2011.

[26] R. Kuntz, J. Montavont, and T. Noël, “Multiple Mobile Routers in NEMO: How Neighbor Discovery Can Assist Default Router Selection”, in *proc. of the IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2008.

[27] J. J. Montavont, E. Ivov, and T. Noël, “Analysis of Mobile IPv6 Handover Optimizations and their Impact on Real-Time Communication”, in *proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2007.

[28] N. Montavont, J. Montavont, and S. Hachana, “Wireless IPv6 Simulator: SimulX”, in *proc. of the ACM/SIGSIM Communications and Networking Simulation Symposium (CNS), located with Spring Simulation Multiconference (SpringSim)*, 2007.

[29] J. Montavont, J. Lorchat, and T. Noël, “Deploying NEMO: A Practical Approach”, in *proc. of the International Conference on ITS Telecommunications (ITST)*, 2006.

[30] J. Montavont and T. Noël, “IEEE 802.11 Handovers Assisted by GPS Information”, in *proc. of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2006.

[31] J. Montavont, N. Montavont, and T. Noël, “Enhanced schemes for L2 handover in IEEE 802.11 networks and their evaluations”, in *proc. of the IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 2005.

## International workshops

[32] S. Sampayo, J. Montavont, and T. Noël, “A Performance Study of the Behavior of the Wake-Up Radio in Real-World Noisy Environments”, in *proc. of the Workshop on Wake-Up radio technologies for next generation wireless communications (AWAKE), in conjunction with EWSN 2020*, 2020.

[33] A. Boutet, B. Le Texier, J. Montavont, N. Montavont, and G. Schreiner, “Advantages of Flow Bindings: an Embedded Mobile Network Use Case”, in *proc. of the Workshop on Experimental Evaluation and Deployment Experiences on Vehicular Networks (WEEDEV), in conjunction with TRIDENTCOM*, 2008.

[34] S. Vincent, J. Montavont, and N. Montavont, “Implementation of an IPv6 Stack for NS-3”, in *proc. of the International Workshop on NS-2 (WNS2), in conjunction with the 3rd International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS)*, 2008.

## National conferences

[35] S. Sampayo, J. Montavont, and T. Noël, “Selecting Parents with Wake-Up Radios for Load Balancing in RPL”, in *actes du colloque francophone sur la conception de protocoles, l'évaluation de performance et l'expérimentation des réseaux de communication (CoRes)*, 2019.

[36] J. Montavont, A. Gallais, and T. Noël, “De l'Internet mobile à la mobilité dans l'Internet des objets”, in *actes des journées francophones Mobilité et Ubiquité (Ubimob), article invité*, 2012.

[37] D. Roth, J. Montavont, and T. Noël, “MOBINET : Gestion de la Mobilite a travers differents Reseaux de Capteurs Sans Fil”, in *actes des Rencontres Francophones sur les Aspects Algorithmiques de Telecommunications (Algotel)*, 2010.

[38] J. Montavont, T. Noël, and K. Guillouard, “Anticipation des Handovers des Nœuds IPv6 a l'aide d'information de Géolocalisation”, in *actes du Colloque Francophone sur l'Ingenierie des Protocoles (CFIP)*, 2006.

## Demonstrations

[39] J. Beaudaux, A. Gallais, R. Kuntz, *et al.*, “CASINO: Creating Alea with a Sensor-based Interactive Network”, in *demonstration abstract proc. of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2010.

# Appendix B

## PhD Supervision

1. **Amaury Bruniaux** (2020 - ...)
   Title: *SDN approach to support ubiquitous IoT technologies*
   Supervisors: Julien montavont (25%), Nicolas Montavont (25%), Georgios Papadopoulos (25%) and Thomas Noël (25%)
2. **Farzad Veisi** (2020 - ...)
   Title: *Software Defined Industrial Internet of Things*
   Supervisors: Julien Montavont (50%) and Fabrice Théoleyre (25%)
3. **Sébastian Lucas Sampayo** (2018 - 2021)
   Title: *Polymorphic network protocol suite in heterogeneous wake-up IoT networks*
   Supervisors: Julien Montavont (50%) and Thomas Noël (50%)
4. **Cosmin Cobârzan** (2012 - 2015)
   Title: *Internet of Highly Mobile Things*
   Supervisors: Julien Montavont (50%) with Thomas Noël (50%)
5. **Damien Roth** (2009 - 2012)
   Title: *Mobility management in Wireless Sensor Networks*
   Supervisors: Julien Montavont (50%) with Thomas Noël (50%)
6. **Romain Kuntz** (2007 - 2010)
   Title: *Medium Access Control Facing the Dynamics of Wireless Sensor Networks*
   Supervisors: Julien Montavont (33%) with Antoine Gallais (33%) and Thomas Noël (33%)

# Appendix C

# Curriculum Vitae

## C.1 Experience and education

- **Assistant professor** (since 2007)
  Mathematics and Computer Science faculty
  ICube lab (UMR CNRS 7357), University of Strasbourg, France
- **Visiting researcher** (2008 - 3 months)
  Polytechnique Montréal, LARIM Lab, Montréal, Canada
- **Ph.D. in Computer Science** (2004 - 2006)
  Title: *Mobility management of IPv6 devices using location systems*
  Advisor: Pr. Thomas Noël
  Louis Pasteur University, Strasbourg, France

## C.2 Research activities

### C.2.1 Academic duties

| | |
|---|---|
| Since 2022 | Deputy head of network research group, ICube lab |
| 2020 | Chair of the selection board of the assistant professor position n°0198/4645, University of Strasbourg, France |
| 2016 - current | Member of the computer science board of the University of Strasbourg, France |
| 2015 | Reviewer in the Ph.D. committee of Hanane El Abdellaouy on *Mécanisme d'optimisation de l'utilisation des technologies dans le Home Network*, under supervision of Dr. Laurent Toutain |
| 2013 - 2017 | Elected member of the computer science dpt, ICube lab, Strasbourg France |
| 2009 - 2012 | Elected member of the computer science dpt, LSIIT lab, Strasbourg France |

### C.2.2 Organization committees

| | |
|---|---|
| 2021 | RESCOM Fall School on Reproducibility in Networking and Systems, Strasbourg France (member) |
| 2020 | 3rd Summer School on Future IoT, IoT Meets Security, Online (member) |
| 2020 | Workshop AWAKE, in conjunction with EWSN, Lyon France (co-chair) |
| 2018 | AdHoc-Now, St Malo France (TPC co-chair) |
| 2016 | IFIP Wireless Days, Toulouse France (track co-chair) |
| 2016 | RESCOM Summer School on 5G and IoT, Guidel France (co-chair) |
| 2011 | RGE days, Strasbourg France (member) |
| 2009 | CFIP Shadow PC, Strasbourg France (co-chair) |
| 2009 | CFIP conference, Strasbourg France (member) |
| 2008 | RESCOM days, Strasbourg France (member) |

### C.2.3 Technical program committees

| | |
|---|---|
| 2022 | IEEE WiMoB, Thessaloniki Greece |
| 2022 | IEEE ISCC, Rhodes Island Greece |
| 2020 | IEEE ISCC, Rennes France |
| 2020 | AdHoc-Now, Bari Italy |
| 2019 | AdHoc-Now, Luxembourg |
| 2011 | NS-3 workshop, in conjunction with SIMUTools, Barcelona Spain |

### C.2.4 External reviewer

| | |
|---|---|
| Journals | Annals of Telecommunications, Elsevier Ad Hoc Networks, Elsevier Communication Networks, Elsevier Computer Communications, Elsevier Physical Communication, IEEE Communications Letters, IEEE Wireless Communications, IEEE Sensors, IEEE Transaction on Vehicular Technology, IET Wireless Sensors Systems, Journal of Distributed Sensor Networks, MDPI Sensors, Springer Wireless Networks, Springer Telecommunication Systems |
| Conferences | IEEE Globecom, IEEE ICC, IEEE ISCC, IEEE IWCMC, IEEE PIMRC, IEEE WiMoB |

### C.2.5 Talks

2014 *Adding values to wireless network evaluation with the FIT IoT-LAB project*, Shanghai Jiao Tong University, China

2013 *Mobility support in wireless sensor networks and the Internet of Things*, Shanghai Jiao Tong University, China

2013 *The FIT IoT project: a very large scale open embedded communicating objects platform in France*, workshop CINTRA (UMI CNRS 3288), Nanyang Technological University, Singapore

2012 *De l'Internet mobile à la mobilité dans l'Internet des Objets*, French days UBIMOB, Anglet France

2012 *La mobilité dans les réseaux de capteurs sans fil*, Wireless Network Workshop, University of Reims France

### C.2.6 Research projects

2018 - 2022 WAKE-UP (French National Research Agency), How wake-up radio solutions will bring new paradigms for heterogeneous M2M networks

2015 - 2016 LoRaCROFT (Inter-Carnot MINES - TSN), Long-range radio: networking next generation communicating objects

2015 FIT-III, Memorandum of Understanding (MoU) between FIT IoT-lab and Institute for Information Industry (Taiwan)

2014 - 2015 UBIQUITY (API ICube), Auto-configurable wireless sensor network for Urban climatology investigation

2012 - 2015 IRIS (French National Research Agency), IP networks for next generation IoT

2011 - 2019 FIT (Investissements d'avenir, Equipex), Open large-scale infrastructure for systems and applications on wireless and sensor communications

2011 QoSMR (CRE Orange Labs), Global QoS for mobile routers

2010 - 2013 DAHLIA (DGCIS/ANSP), Experimental platform of home automation for telehomecare and remote monitoring for elderly

2010 EXPRIMA (BQR University of Strasbourg), Experimentations for wireless mesh networks

2008 - 2011 SensLAB (French National Research Agency), Very large-scale open wireless sensor network testbed

2006 - 2008 REMORA (French National Research Agency), Mobility support for autonomous networks

2006 - 2008 SINEMA (CRE Orange labs), Service discovery in mobile environment

## C.3 Teaching activities

### C.3.1 Coordinator (degree) - University of Strasbourg

| Period | Diploma | Year | Rate |
|---|---|---|---|
| 2019 - 2021 | Engineering prog. in Computer Science, Systems and Networking | 4 and 5 | 50% |
| 2018 - 2021 | M.Sc. in Computer Science, networking and Systems | 1 and 2 | 50% |
| 2015 - 2019 | M.Sc. in Networking and Embedded Systems | 2 | 100% |
| 2008 - 2014 | Bachelor in Computer Network Management (apprenticeship program) | 3 | 100% |

### C.3.2 Head teacher (course)

| Period | Course | Location | ECTS | Level |
|---|---|---|---|---|
| 2019 - 2022 | Wireless Networks | Faculty of Maths and C.Sc. | 3 | M.Sc. |
| 2018 - 2022 | IP Networks | Faculty of Maths and C.Sc. | 3 | B.Sc. |
| 2017 - 2022 | System Programming | Telecom Physics Strasbourg | 3 | B.Sc. |
| 2016 - 2022 | System Programming | Faculty of Maths and C.Sc. | 3 | B.Sc. |
| 2015 - 2018 | Local Networks | Faculty of Maths and C.Sc. | 3 | B.Sc. |
| 2013 - 2019 | Internet of Things | Faculty of Maths and C.Sc. | 3 | M.Sc. |
| 2012 - 2019 | Multimedia and QoS | Faculty of Maths and C.Sc. | 6 | M.Sc. |
| 2012 - 2013 | Embedded OS | Faculty of Maths and C.Sc. | 3 | M.Sc. |
| 2009 - 2012 | System and Internet Admin | IUT Robert Schuman | 3 | B.Sc. |
| 2008 - 2018 | Networks and Protocols | Faculty of Maths and C.Sc. | 6 | B.Sc. |
| 2008 - 2018 | Applications and Services | IUT Robert Schuman | 6 | B.Sc. |
| 2008 - 2013 | Prototyping and Simulating | Faculty of Maths and C.Sc. | 3 | M.Sc. |
| 2007 - 2012 | Deploying OS and Networks | IUT Robert Schuman | 6 | B.Sc. |
| 2007 - 2012 | Using OS and Networks | IUT Robert Schuman | 6 | B.Sc. |

### C.3.3 Teacher

| Period | Course | Location | ECTS | Level |
|---|---|---|---|---|
| 2021 - 2022 | OS Architecture | Faculty of Maths and C.Sc. | 6 | B.Sc. |
| 2019 - 2022 | Programmable Networks | Faculty of Maths and C.Sc. | 3 | M.Sc. |
| 2020 - 2021 | Network Algorithms | Faculty of Maths and C.Sc. | 3 | B.Sc. |
| 2020 - 2021 | Computer Architecture | Faculty of Maths and C.Sc. | 3 | B.Sc. |
| 2018 - 2022 | Operating Systems | Telecom Physics Strasbourg | 3 | B.Sc. |
| 2011 - 2012 | Distributed Systems | Faculty of Maths and C.Sc. | 3 | B.Sc. |

### C.3.4 Internship and project supervision (per-year)

| Type | Duration | Level | Load |
|---|---|---|---|
| Internship | 6 months | M.Sc. (CMI year 5) | 2 students |
| Internship | 6 months | M.Sc. | 3 students |
| Internship | 6 months | Engineering program | 2 students |
| Internship | 3 months | M.Sc. (CMI year 4) | 2 students |
| Apprenticeship | 1 year | B.Sc. | 1 student |
| Internship | 1 month | B.Sc. | 1 student |
| Final year project | 150 hours | M.Sc. | 1 group |
| Research project | 150 hours | M.Sc. | 3 students |
| Scientific documentation project | 60 hours | B.Sc. | 1 group |

### C.3.5 Teaching hours

Figure C.1 presents the number of teaching hours I provided per year. The orange line shows the statutory value (set to 192 hours per year).

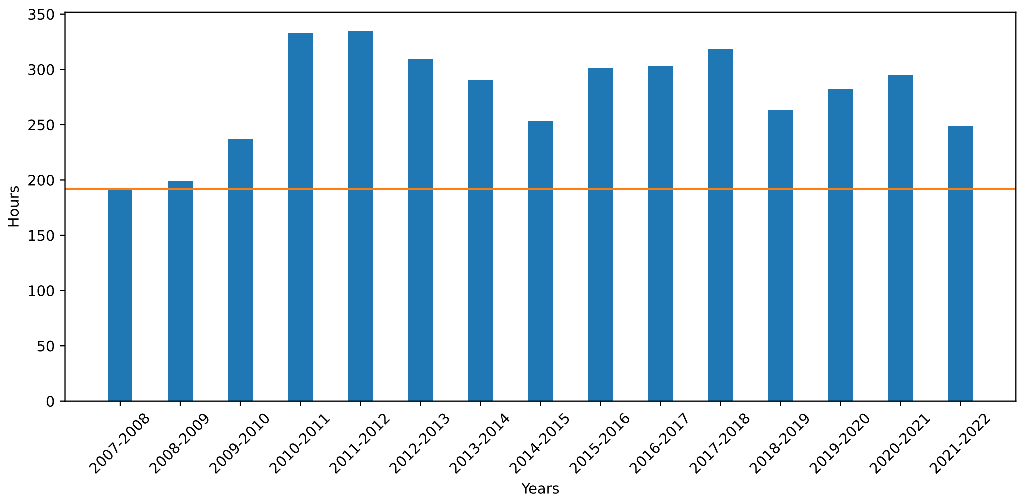



**Figure C.1:** Number of teaching hours per year

### C.3.6 Extra activities

2014 Course on IP version 6, *Ayitic - Internet pour le développment* program, founded by the Latin American and Caribbean Internet Address Registry (LACNIC), Port-au-Prince Haïti

2011 - 2013 Computer science training for high school teachers, in anticipation of the optional computer science course (ISN) opening in final year of high school

### C.3.7 Publication

S. Cateloin, A. Gallais, S. Marc-Zwecker and J. Montavont, *Mini-manuel des réseaux informatiques*, Dunod, 2012

# Appendix D

## Version française abrégée

### D.1 Introduction

#### D.1.1 Contexte

Les avancées réalisées dans les circuits intégrés ont permis de combiner, sur un même équipement, des capteurs, actionneurs et des transmetteurs radio, afin de collecter des données physiques ou d'agir sur un système pour en modifier son état. Ces nouveaux objets peuvent être mis en réseau via des communications sans fil afin d'automatiser ou de centraliser certaines tâches. Les réseaux de communication formés par ces objets communicants sont caractérisés par des ressources énergétiques réduites, des portées radios limitées, des débits faibles et des liens instables pouvant générer des pertes. Dans la suite de ce document, nous nommerons ces réseaux *Low-power and Lossy Wireless Networks* (LLWN).

En raison de la faible portée radio des équipements, il peut être nécessaire de relayer les données de proche en proche afin d'atteindre une destination éloignée d'une source. Dès lors, les nœuds d'un réseau LLWN jouent le rôle de routeurs afin d'assurer cette tâche, en plus de la génération de données. Une telle organisation est illustrée sur la figure D.1 sur laquelle le nœud $S$ peut joindre le nœud $P$ à l'aide de deux relais intermédiaires. Les réseaux LLWN sont généralement utilisés pour véhiculer un trafic convergent, c'est-à-dire que tout le trafic issu des nœuds est transmis à une destination unique. Ce nœud spécial peut alors stocker localement les données reçues, ou alors servir de passerelle vers un réseau de distribution (e.g. Internet) afin d'acheminer les données collectées vers des serveurs distants. Sur la figure D.1 le nœud $P$ joue ce rôle de passerelle, permettant au nœud $S$ de transmettre ses données au nœud $D$ alors que ce dernier se trouve en-dehors du réseau LLWN. Cette organisation forme ce qu'on appelle l'Internet des Objets (IoT).

#### D.1.2 Cas d'étude

Le support de la mobilité, c'est-à-dire la possibilité de communiquer tout en se déplaçant, est une fonctionnalité qui permet d'étendre l'usage des réseaux LLWN à de nombreux domaines, tels que le suivi logistique, la santé, les transports ou l'armée [4]. Malheureusement, la présence de nœuds mobiles peut perturber le fonctionnement du réseau et avoir des répercussions

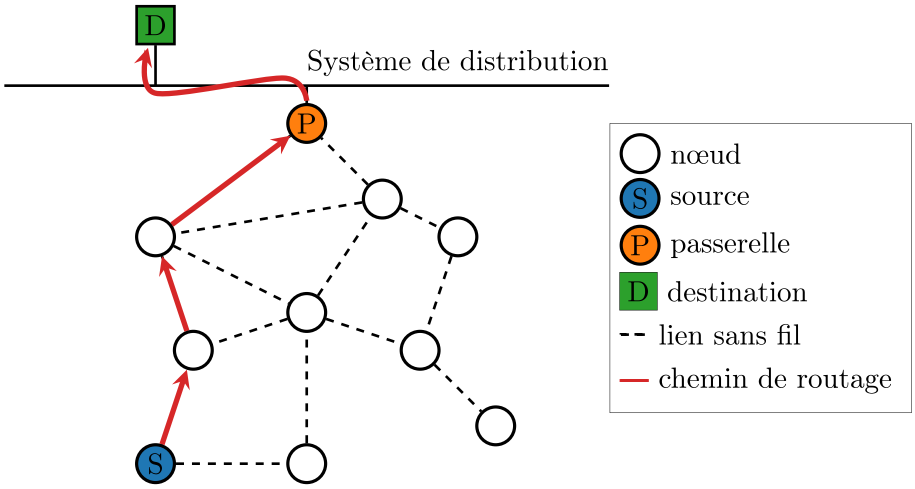



**Figure D.1:** Exemple d'un réseau sans fil contraint et multi-saut (LLWN)

sur la qualité des communications, les nœuds mobiles pouvant se retrouver déconnectés du réseau pendant de longues périodes. Une telle situation s'explique principalement par des raisons historiques : les paradigmes et abstractions mis en œuvre encore aujourd'hui dans les réseaux LLWN ont été définis il y a plus de 40 ans pour des réseaux de communication filaires dans lesquels aucune mobilité n'était possible.

Le support de la mobilité doit permettre aux nœuds mobiles d'insérer leurs transmissions dans l'ordonnancement réalisé pour gérer l'accès au support de communication. Cette opération représente un défi car l'environnement proche d'un nœud mobile change en fonction de sa position dans le réseau, ce qui peut introduire des problèmes de synchronisation. Ce problème est étudié dans la section D.2. Les nœuds mobiles peuvent également être destinataires d'une communication, ce qui complexifie encore le support de la mobilité car le déploiement de routes descendantes fiables (depuis un correspondant vers un nœud du réseau LLWN) reste un verrou scientifique [16] encore non résolu. Une telle situation nécessite des mécanismes supplémentaires au niveau de la couche réseau et nous présentons nos contributions sur ce thème dans la section D.3. Nous avons également cherché à repousser les limites des protocoles asynchrones à travers l'utilisation des Wake-Up radios. Nos contributions au niveau MAC et réseau sont présentées dans la section D.4. Enfin, la conclusion de nos travaux et notre projet de recherche pour les prochaines années sont présentés dans la section D.5.

### D.1.3 Hypothèses de travail et méthodologie

Le travail présenté dans ce document utilise les hypothèses de travail suivantes :

- les réseaux LLWN sont composés de nœuds mobiles et fixes;
- les nœuds sont conscients de leur nature (mobile ou fixe), ce qui permet la définition de comportements spécifiques en fonction du type des nœuds ;

- l'accès au support de communication est réalisé via un protocole asynchrone. Les méthodes synchronisées sont actuellement plébiscitées par la communauté scientifique mais nous démontrerons dans la section D.4 qu'un protocole asynchrone peut offrir des performances équivalentes voire supérieures. De plus, un protocole asynchrone s'accommode mieux d'un environnement dynamique ;
- les nœuds mobiles ne relaient pas de trafic afin d'éviter le calcul et le déploiement de routes instables.

Nos contributions sont validées par simulations avec les simulateurs WSNet et Cooja reconnus par la communauté scientifique. Certains résultats sont consolidés par des expérimentations, soit en utilisant un déploiement ad hoc de quelques nœuds, soit en utilisant la plateforme FIT IoT-LAB dès que possible afin de rendre nos résultats reproductibles.

## D.2 Support de la mobilité au niveau MAC

Les travaux présentés dans cette section sont issus des travaux de thèse de M. Romain Kuntz (2007 - 2010) et de M. Damien Roth (2009 - 2012), respectivement co-encadrés avec les professeurs Antoine Gallais et Thomas Noël, et Thomas Noël.

### D.2.1 Contexte de recherche

Lorsque le support de communication est partagé entre plusieurs nœuds, un protocole MAC est nécessaire pour ordonnancer l'accès à ce support afin d'éviter les collisions et par conséquent le gaspillage des ressources. Afin de limiter la consommation énergétique des nœuds, le protocole MAC peut soumettre le transmetteur radio à un cycle de travail, alternant les périodes actives (le transmetteur peut recevoir ou envoyer des octets) et de veille (le transmetteur ne peut ni recevoir ni émettre des octets). De nombreux protocoles MAC ont été proposés dans la littérature [19] et peuvent être catégorisés en trois familles de protocoles : asynchrones, synchronisés et hybrides. Nous nous sommes focalisés sur la première famille car elle s'accommode mieux d'un environnement dynamique tel que celui défini par la présence de nœuds mobiles.

Les protocoles asynchrones permettent aux différents nœuds de gérer leurs cycles de travail en autonomie c'est-à-dire sans synchronisation globale. Chaque transmission de données est précédée par l'envoi d'un préambule dont la durée d'émission est supérieure à la durée de veille définie dans le cycle de travail. Lors de son prochain réveil, le récepteur détectera le préambule et ainsi restera actif afin de recevoir les données transmises à la fin du préambule. S'il n'y a pas d'activité sur le support de communication au réveil d'un nœud, ce dernier repasse directement en veille jusqu'à la prochaine période d'activité dictée par le cycle de travail. Ce mode de fonctionnement est communément appelé *Low Power Listening* (LPL) dans la littérature.

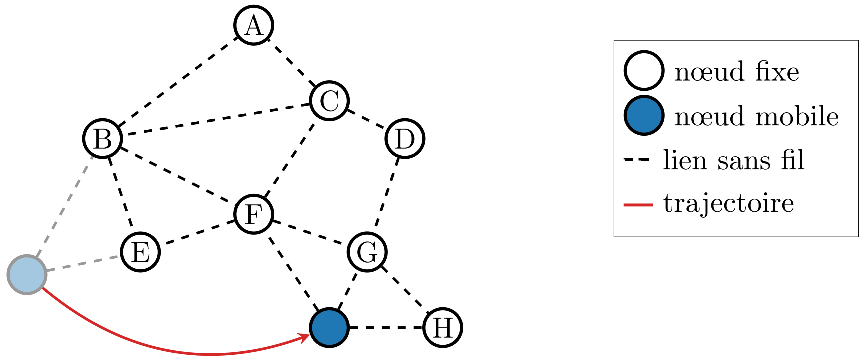



**Figure D.2:** Un nœud mobile se déplaçant au sein d'un réseau LLWN

Nous définirons la mobilité comme le fait de communiquer (échanger des informations) tout en se déplaçant. En d'autres termes, le voisinage d'un nœud mobile risque de changer en fonction de sa position, comme cela est illustré sur la figure D.2. Au départ, le nœud mobile est à portée radio des nœuds $B$ et $E$. Après son déplacement, ses nouveaux voisins sont les nœuds $G$, $H$, et $F$ alors que $B$ et $E$ sont désormais hors de portée.

Dans cette situation, les communications courantes du nœud mobile doivent être maintenues indépendamment de ses déplacements. Grâce à l'utilisation d'un protocole MAC asynchrone, le nœud mobile peut concourir comme n'importe quel autre nœud pour accéder au support de communication. Néanmoins, la résolution du prochain saut, c'est-à-dire vers quel nœud intermédiaire à portée radio transmettre le message, reste à déterminer. Nous avons montré qu'une stratégie à base de transmissions *broadcast* (tous les nœuds à portée radio acceptent le message) n'est pas adaptée car elle augmente le taux de paquets perdus [26]. Ce résultat s'explique par des problèmes de synchronisation liés aux déplacements des nœuds mobiles : des données sont envoyées alors que l'ensemble des voisins sont en veille car ils n'ont pas eu l'opportunité d'entendre le préambule. L'absence d'accusés de réception pour les transmissions *broadcast* affecte également le taux de pertes. De plus, une forte compétition pour accéder au support de communication augmente les délais de transmission, ce qui accentue les problèmes de synchronisation et augmente le cycle de travail ayant pour conséquence une consommation énergétique accrue. Au regard de ces résultats, nous avons préconisé l'utilisation de transmissions unicast et abordé la détermination du prochain saut avec deux approches qui sont détaillées dans les sections suivantes.

### D.2.2 Contributions

#### Construction d'une liste de voisins

Afin d'utiliser des transmissions unicast, les nœuds mobiles doivent déterminer un prochain saut valide vers qui transmettre leurs données. Pour ce faire, nous proposons que les nœuds mobiles construisent et maintiennent à jour une liste des voisins joignables à l'aide d'écoutes passives du support de communication [27]. Les voisins actifs peuvent être identifiés à l'aide des informations de contrôle véhiculées dans les entêtes des messages. Pour limiter la consommation énergétique introduite par ces écoutes passives, nous avons défini différents déclencheurs qui permettent d'initier une phase d'écoute. En particulier, le déclencheur *duty-cycle* réalise des écoutes passives de manière périodique.

Nous avons implémenté cette solution dans le simulateur WSNet et comparé les performances obtenues avec une stratégie qui utilise des transmissions broadcast. L'ensemble des paramètres de simulation est disponible dans [27]. La figure D.3 représente le trafic transmis par les nœuds mobiles et reçu par la passerelle. On peut voir que l'utilisation de transmissions broadcast génère de nombreuses duplications, ce qui augmente la charge du réseau et par conséquent augmente la consommation d'énergie sur les nœuds fixes de 15%. À l'inverse, nos écoutes passives évitent un tel phénomène, mais le taux de messages perdus augmente de 14% et le coût énergétique est ici imputé aux nœuds mobiles. Néanmoins, le déclencheur *duty-cycle* couplé avec des durées de veille courtes permet un taux de correspondance relativement élevé (dans 60% des cas la liste des voisins est à jour et permet une transmission unicast) tout en limitant le surcoût énergétique lié à notre solution.

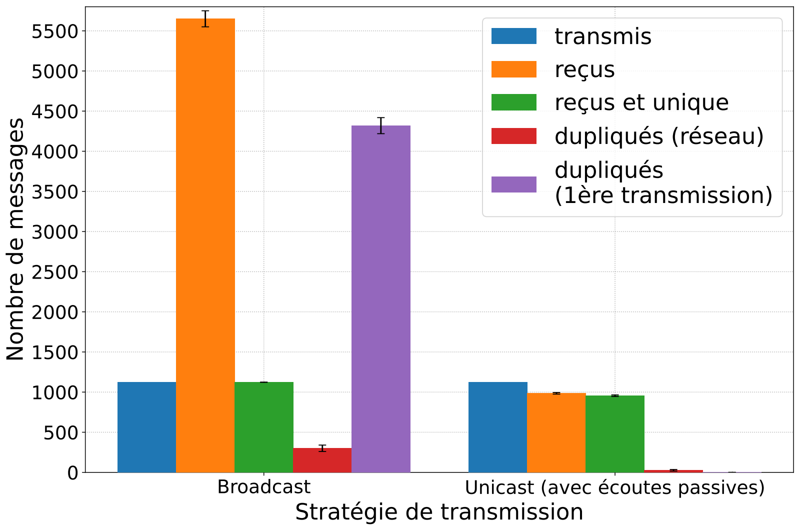



**Figure D.3:** Trafic issu des nœuds mobiles

Dans cette solution, les nœuds mobiles restent en compétition comme tout autre nœud pour accéder au support de communication. Par conséquent, des problèmes de synchronisation peuvent se produire lorsque le prochain saut sélectionné en début de transmission n'est plus joignable lors de la transmission effective des données. Pour limiter ce phénomène, il est possible de rompre l'équité dans l'accès au support de communication en favorisant les nœuds mobiles. Cette solution constitue notre seconde contribution.

**Favoriser les nœuds mobiles dans l'accès au support de communication**

Notre contribution, appelée X-Machiavel, est basée sur le protocole X-MAC [29] et bénéficie des préambules fragmentés introduits par ce dernier. Notre solution permet aux nœuds mobiles de bénéficier de relais fixes opportunistes lorsque le support de communication est libre, ou de subtiliser le support lorsqu'il est utilisé par un nœud fixe. Dans le premier cas, les nœuds mobiles utilisent un type de préambule spécifique qui offre la possibilité aux nœuds fixes du voisinage de se déclarer comme destinataires intermédiaires des données à venir. Dans le second cas, un nœud mobile peut directement transmettre ses données entre deux fragments de préambule au nœud fixe qui utilise actuellement le support de communication.

Cette solution a été évaluée par simulation sur le simulateur WSNet. L'ensemble des paramètres de simulation est disponible dans [31]. La figure D.4 présente le taux moyen de pertes au niveau MAC et les raisons de ces pertes. Comme nous pouvons le voir sur cette figure, X-Machiavel réduit le taux de pertes de 40% par rapport à X-MAC. Ce résultat s'explique principalement par le fait que X-Machiavel limite les problèmes de synchronisation entre un nœud mobile source et son prochain saut fixe. En réduisant le taux de pertes, X-Machiavel permet dans le même temps de réduire la consommation énergétique de l'ensemble du réseau, comme illustré par la figure D.5. Enfin, X-Machiavel réduit par un facteur 2 le délai nécessaire pour accéder au support de communication pour les nœuds mobiles. Cela permet de réduire la compétition pour accéder au support de communication, et par conséquent réduit également le délai subit par les nœuds fixes.

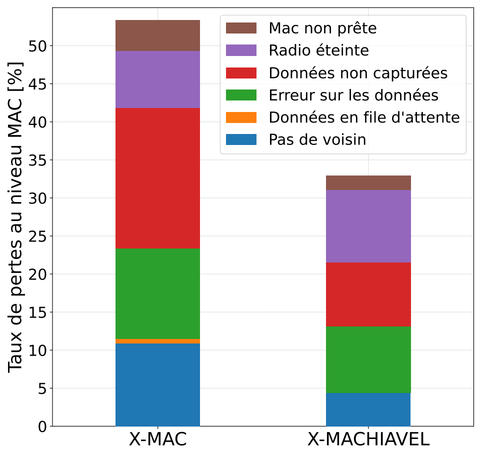



**Figure D.4:** Pertes au niveau MAC

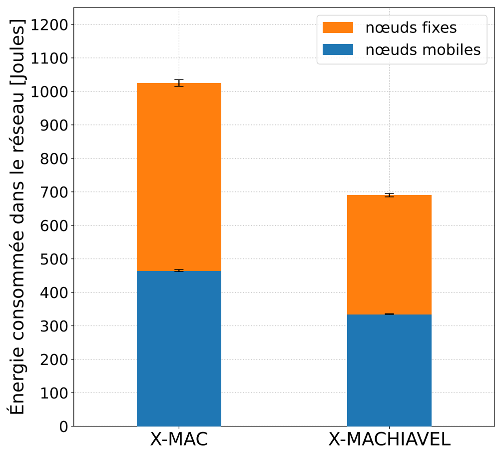



**Figure D.5:** Consom. énergétique

### D.2.3 Conclusion et perspectives

Dans cette section, nous avons étudié en quoi la mobilité affecte les communications au niveau MAC, c'est-à-dire au niveau de la couche liaison de données du modèle OSI. Les principaux défis consistent à identifier un prochain saut valide afin d'utiliser des communications unicast et de minimiser le temps de transmission des nœuds mobiles afin de limiter les défauts de synchronisation. Nous avons proposé l'utilisation d'écoutes passives afin que les nœuds mobiles maintiennent une liste des voisins actifs vers qui envoyer leurs données. Cependant, nous avons toujours observé des problèmes de synchronisation entre émetteur et récepteur. C'est pourquoi nous avons proposé le protocole X-Machiavel qui permet aux nœuds mobiles de bénéficier de relais opportunistes ou de voler le support de communication si pré-réservé par un nœud fixe. De mon point de vue, cette seconde contribution répond de manière satisfaisante aux deux défis suscités.

Depuis nos contributions, la majorité des solutions MAC pour offrir un support de la mobilité dans les réseaux LLWN reprend les concepts définis dans X-Machiavel comme dans le protocole MOX-MAC [36], ou les adapte à des situations particulières comme dans le protocole MobiXplore [40] qui cherche à optimiser la gestion des rafales de trafic en présence de nœuds mobiles. Nous avons néanmoins identifié deux pistes de recherche qui nous semblent prometteuses. Dans un premier temps, nous aimerions étudier comment la technologie Wake-Up radio pourrait améliorer les performances au niveau MAC et le support de la mobilité en particulier. Nous avons déjà débuté cette étude et nos premiers résultats sont présentés dans la section D.4. En parallèle, nous pouvons changer le paradigme de communication et considérer les protocoles MAC orientés récepteurs tels que RI-MAC [43] qui résolvent par leur concept même l'identification d'un prochain saut. Néanmoins, cette famille de protocoles présente des problèmes de consommation énergétique qui sont principalement liés aux longues périodes d'écoute passive. La gestion des collisions entre plusieurs récepteurs ainsi que le support des communications broadcast restent également des défis ouverts.

## D.3 Support de la mobilité au niveau réseau

Les travaux présentés dans cette section sont issus des travaux de thèse de M. Damien Roth (2009 -2012) et M. Cosmin Cobârzan (2012 - 2015), co-encadrés avec le professeur Thomas Noël.

### D.3.1 Contexte de recherche

Dans la section précédente nous avons émis l'hypothèse que les nœuds mobiles étaient uniquement producteurs de données. Nous considérons ici que les nœuds mobiles sont également consommateurs de données, c'est-à-dire qu'ils peuvent désormais être les destinataires des flux échangés. De nombreuses applications industrielles ou véhiculaires nécessitent l'établissement de communications bidirectionnelles avec des nœuds mobiles [46], [47].

Acheminer des messages jusqu'à un nœud mobile nécessite de connaître sa position réseau. Cette information peut être récupérée depuis son adresse réseau dans le cas où l'espace de

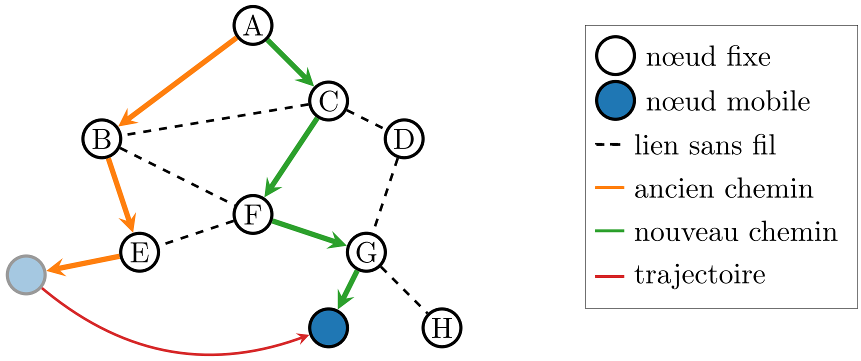



**Figure D.6:** Acheminement de données à un nœud mobile se déplaçant dans un réseau LLWN

nommage utilisé exprime simultanément l'identité et la position des entités, ou depuis le plan de données si le protocole de routage a convergé et que les routes sont à jour. Sur la figure D.6, $A$ souhaite transmettre des données au nœud mobile qui est directement connecté à $E$. Le nœud mobile dispose alors d'une adresse qui indique sa connectivité avec $E$, ou l'ensemble des nœuds sont informés (via le plan de contrôle du protocole de routage) que le nœud mobile est joignable via $E$. Lorsque le nœud mobile se déplace vers $G$, il doit mettre à jour son adresse pour retrouver une connectivité. Cependant, modifier son adresse réseau coupe toutes les communications courantes car cette dernière est utilisée dans les identifiants de session. Il faut également notifier la source (ici $A$) du changement d'adresse du destinataire. Mobile IPv6 est le standard de l'Internet Engineering Task Force (IETF) pour gérer cette situation. Il permet de masquer la position courante des nœuds mobiles via l'utilisation d'un agent relais qui concentre les communications et retransmet les données destinées aux nœuds mobiles vers leurs positions courantes à l'aide de tunnels. Néanmoins nous avons montré que par défaut les nœuds mobiles détectent trop tardivement le changement de connectivité, ce qui augmente le temps de déconnexion et le taux de paquets perdus [55].

L'alternative nécessite de déclencher une mise à jour du plan de données et d'attendre que le protocole de routage converge à nouveau. Dans ce scénario, nous avons étudié *IPv6 Routing Protocol for Low-Power and Lossy Networks* (RPL) [10] qui est l'un des protocoles de routage préconisé par l'IETF pour les réseaux LLWN. Malheureusement, RPL présente des temps de convergence non négligeables et nous avons montré que les mises à jour sont déclenchées trop tardivement et contribuent à l'augmentation du temps de déconnexion [56]. Au regard de ces résultats, nous avons étendu nos travaux précédents afin de minimiser le temps de déconnexion réseau des nœuds mobiles, aussi bien avec Mobile IPv6 qu'avec RPL. Ces contributions sont présentées dans la section suivante.

### D.3.2 Contributions

#### Minimiser le temps de déconnexion dans Mobile IPv6

La principale limitation du protocole Mobile IPv6 provient de la détection de mouvement, c'est-à-dire le mécanisme utilisé pour détecter le changement de position réseau et ainsi déclencher la mise à jour de l'agent relais [52]. Pour réduire ce temps de détection, nous avons proposé d'étendre nos travaux précédents sur les écoutes passives. Lorsque la liste de voisins construite par un nœud mobile change significativement, cela déclenche une vérification de la connectivité réseau courante. Si le nœud mobile reçoit une confirmation de son ancien routeur, il en déduit que sa position réseau n'a pas changé. À l'inverse, si la confirmation provient d'un nouveau routeur, le nœud mobile déclenche les opérations liées à Mobile IPv6 (création d'une nouvelle adresse, notification de l'agent relais et mise à jour des tunnels). Cette contribution est référencée dans la suite comme *mobinet*.

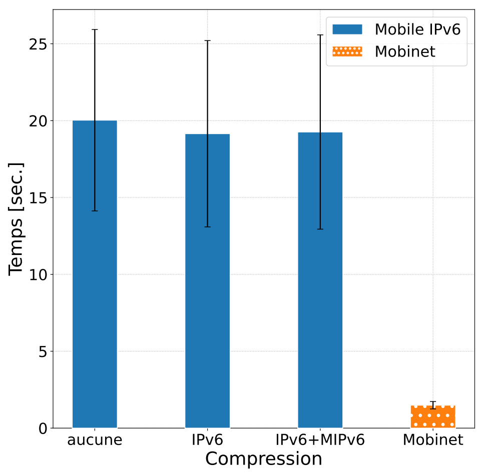



**Figure D.7:** Durée totale de déconnexion

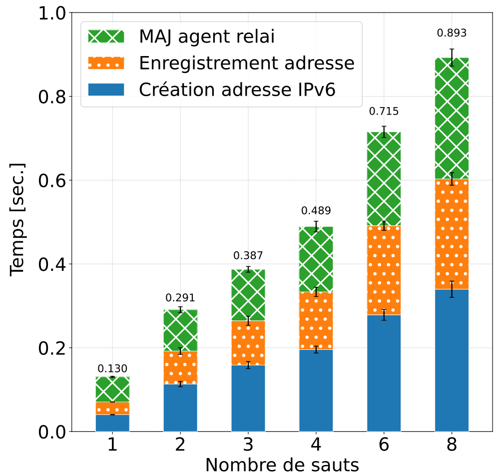



**Figure D.8:** Délais liés à Mobile IPv6

Cette contribution a été évaluée par expérimentations sur la plateforme FIT IoT-LAB. La configuration utilisée est détaillée dans [55] et [58]. La figure D.7 présente les temps de déconnexion d'un nœud mobile se déplaçant dans un réseau LLWN. Mobinet permet de limiter le temps de déconnexion à 1,5 sec. en moyenne, alors que sans optimisation le temps de déconnexion est directement lié à l'expiration d'un temporisateur (ici fixé à 30 sec., mais dont la valeur peut atteindre 18 heures). On notera que la compression des entêtes IPv6 (avec [11]), ou IPv6 et Mobile IPv6 (avec [11] et [59]) a une influence limitée sur le temps de déconnexion, mais permet néanmoins un gain d'environ 1 sec. Passée la détection de mouvement, Mobile IPv6 nécessite environ 130 ms pour créer une nouvelle adresse, l'enregistrer auprès du routeur et mettre à jour l'agent relais comme illustré sur la figure D.8. Mobinet n'ayant pas d'influence sur ces opérations, les résultats sont similaires entre notre contribution et la version native de Mobile IPv6. Augmenter la distance réseau (en nombre de sauts) entre le nœud mobile et l'agent relais augmente linéairement le temps nécessaire pour réaliser ces opérations, chaque saut supplémentaire ajoutant environ 150 ms jusqu'à atteindre environ 900 ms pour 8 sauts, ce qui correspond à un déploiement à très large échelle.

À travers ces travaux nous avons démontré que Mobile IPv6 peut être une solution viable pour gérer la mobilité au niveau réseau dans les réseaux LLWN. Grâce à nos écoutes passives, nous avons été en mesure de réduire significativement le temps de déconnexion qui est principalement lié à la détection de mouvement. Néanmoins, cette solution n'est pas entièrement satisfaisante : elle introduit un point unique de défaillance (via l'agent relais) et peut engendrer un routage sous-optimal. Pour toutes ces raisons, j'ai décidé d'abandonner cette piste de recherche pour me concentrer sur l'implication du protocole de routage dans la gestion de la mobilité réseau.

### Minimiser le temps de déconnexion dans RPL

Le routage des données dans un réseau LLWN peut être réalisé par le protocole RPL [10]. Ce dernier construit un graphe orienté acyclique dirigé vers une unique destination (DODAG). Lorsque RPL a convergé, chaque nœud (excepté la destination unique) est connecté à au moins un parent. La connectivité avec ce parent peut être surveillée via la réception de messages *DODAG Information Object* (DIO) mais la période d'émission de ces derniers varie en fonction de la stabilité du réseau [61]. Dans ce contexte, nous avons proposé d'inverser le fonctionnement de l'algorithme qui dicte l'émission des DIO sur les parents qui accueillent des nœuds mobiles afin d'anticiper la réception des DIO et de pouvoir réagir en cas de non réception. Nous avons fixé le temporisateur *Dthresh* qui est réinitialisé à chaque réception de DIO. En cas d'expiration de ce temporisateur (des messages DIO n'ont pas été reçus), les nœuds mobiles considèrent qu'ils sont déconnectés du graphe et initient un changement de parent en suivant la procédure définie dans RPL pour effectuer une réparation locale.

Cette contribution, nommée *reverse-trickle* dans la suite, a été évaluée par simulation sur le simulateur WSNet et comparée avec les propositions *PeriodicDIO* [64] et *DynamicDIS* [60]. Nous avons défini deux configurations *reverse-trickle 1* et *reverse-trickle 2* qui diffèrent par les bornes des intervalles dans lesquels *Dthresh* est choisi. L'ensemble des paramètres de simulation est disponible dans [63]. La figure D.9 présente les durées de déconnexion subies par les nœuds mobiles en fonction de la solution utilisée. L'utilisation des solutions *DynamicDIS* et de *PeriodicDIO* nécessitent la réception de nombreux messages DIO avant que la métrique de routage ne converge et permette aux nœuds mobiles de changer de parent, ce qui explique les temps de déconnexion relativement élevés (jusqu'à 102 sec.). La réactivité de notre solution est indépendante de la métrique de routage utilisée et permet aux nœuds mobiles de réagir dès l'expiration du temporisateur *Dthresh*. Par conséquent, la durée de déconnexion est bornée par *Dthresh*. D'autre part, notre solution est uniquement activée sur les parents qui accueillent des nœuds mobiles. Elle est donc plus localisée que les solutions

*DynamicDIS* et *PeriodicDIO* comme on peut le voir sur la figure D.10 qui présente le nombre de messages DIO transmis par l'ensemble des nœuds du réseau.

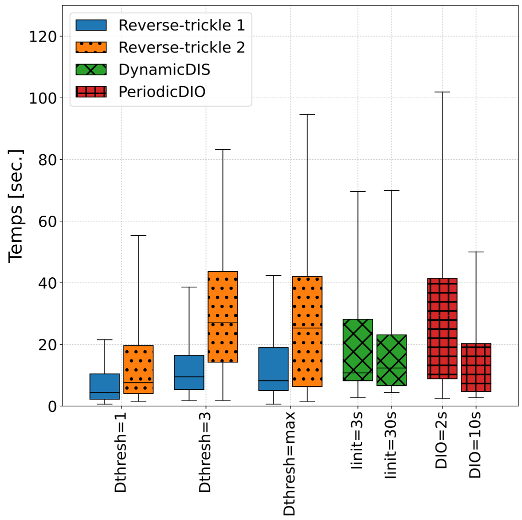



**Figure D.9:** Temps de déconnexion

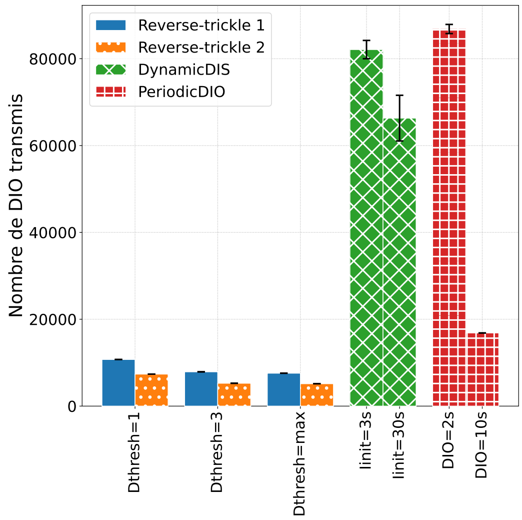



**Figure D.10:** Coût en messages

À la lumière de ces résultats, nous préconisons la configuration 1 et $Dthresh = 3$ car elle présente le meilleur compromis entre réactivité et coût en messages de contrôle. Néanmoins, je reste convaincu qu'une solution qui ne tient compte que d'événements réseaux ne peut pas réagir de manière optimale aux changements qui affectent les couches basses. En renforçant la collaboration entre les couches MAC et réseau je pense qu'il est possible d'améliorer les performances, comme illustré dans la section suivante.

### MT-RPL: une solution inter-couches

Nous proposons ici d'utiliser des événements se produisant au niveau MAC pour déclencher des actions au niveau réseau. Pour ce faire, nous avons étendu notre précédente contribution X-Machiavel afin de déclencher une mise à jour du plan de données dans RPL dès qu'un nœud mobile bénéficie d'un relais opportuniste ou qu'il réussit à subtiliser le canal. En outre, nous avons ajouté le rang RPL dans les préambules envoyés avant chaque transmission de données afin de réserver les actions de X-Machiavel aux nœuds qui permettent de faire progresser les nœuds mobiles dans le DODAG formé par RPL. En l'absence d'acquittement pour un préambule donné, le nœud mobile se détache du DODAG ce qui transforme l'ensemble de ses voisins en parents potentiels. Nous avons nommé cette contribution *MT-RPL*.

Cette contribution a été évaluée par simulation avec le simulateur WSNet et par expérimentation sur la plateforme FIT IoT-LAB. Nous l'avons comparé aux mécanismes suggérés par RPL pour détecter un parent injoignable, à savoir *Neighbor Unreachability Detection* (NUD) [53] et *Bidirectional Forwarding Detection* (BFD) [66]. L'ensemble des paramètres de simulation et d'expérimentation est disponible dans [63] et [56]. La figure D.11 présente les durées de déconnexion pour chaque configuration. Bien que les mécanismes NUD et BFD permettent de réduire la durée de déconnexion, MT-RPL permet de la borner à 20 sec. dans les pires cas (lorsqu'un nœud mobile se détache volontairement du DODAG). Avec NUD et BFD cette performance s'obtient au prix d'un grand nombre de messages de contrôle, comme on peut le voir sur la figure D.12. À l'inverse, MT-RPL offre de meilleures performances sans introduire de nouveaux messages de contrôle. Néanmoins, notre solution s'avère trop sensible et provoque des oscillations dans le réseau. Par conséquent, le nombre de mises à jour du plan de données est exagérément élevé, ce qui explique la contre-performance observé pour MT-RPL sur la figure D.12 concernant les résultats de simulation. Cette observation est corroborée par les taux de livraison présentés dans le tableau D.1. Bien que MT-RPL offre des durées de déconnexion réduite, les oscillations rendent les routes descendantes (depuis la racine du DODAG vers les nœuds mobiles) instables ce qui augmente significativement le taux de pertes. On pourra d'ailleurs remarquer qu'aucune solution ne parvient à maintenir des routes descendantes fiables.

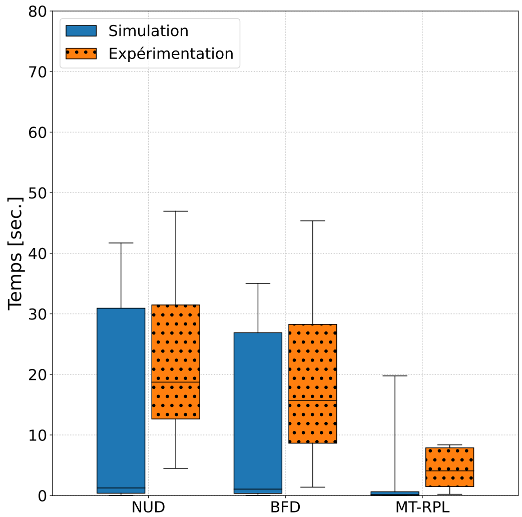



**Figure D.11:** Durée de déconnexion

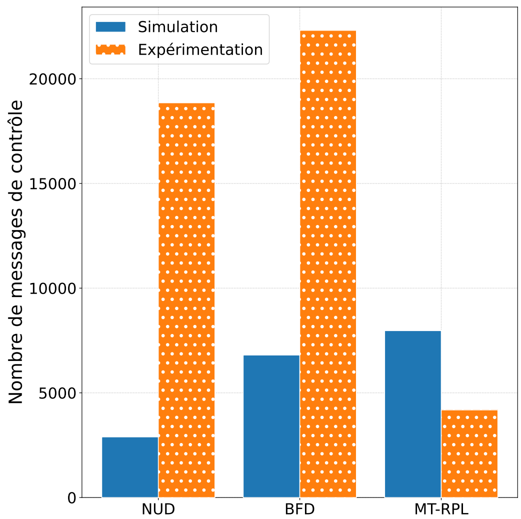



**Figure D.12:** Coût en messages

| Solution | Flux montants | | Flux descendants | |
|---|---|---|---|---|
| | simulation | exp. | simulation | exp. |
| NUD | $10,1\%(\pm 6,6)$ | $87,9\%(\pm 3,4)$ | $8,2\%(\pm 7,4)$ | $28,6\%(\pm 5,4)$ |
| BFD | $18,0\%(\pm 4,5)$ | $58,8\%(\pm 3,7)$ | $14.0\%(\pm 8.0)$ | $26,0\%(\pm 1,9)$ |
| MT-RPL | $62,1\%(\pm 14.0)$ | $94,7\%(\pm 3,1)$ | $23,2\%(\pm 7,6)$ | $19,3\%(\pm 4,6)$ |

**Tableau D.1:** Taux de livraison

### D.3.3 Conclusion et perspectives

Dans cette section, nous avons étudié en quoi la mobilité affecte les communications descendantes, c'est-à-dire lorsque les nœuds mobiles sont consommateurs de données. Le principal défi consiste à déterminer la position réseau des nœuds mobiles. Pour ce faire, nous avons étudié deux alternatives : 1) masquer la position courante des nœuds mobiles par l'utilisation d'un agent relais et de tunnels ; 2) expliciter la position courante des nœuds mobiles avec des mises à jour du plan de données défini par le protocole de routage. Dans les deux cas c'est la détection de mouvement, c'est-à-dire la détection de la perte de connectivité avec son point d'attachement courant que nous avons cherché à optimiser afin de réduire les temps de déconnexion. Nos investigations m'ont amené à abandonner la première solution au profit de la seconde en raison de contraintes rédhibitoires telles que l'introduction d'un point unique de défaillance ou un routage sous-optimal. Notre dernière contribution, MT-RPL, démontre qu'une coopération plus forte entre les couches MAC et réseau permet une meilleure intégration des nœuds mobiles. Néanmoins, elle provoque des changements de parent superflus ce qui génère des oscillations dans le réseau et rend les routes descendantes instables.

De manière générale, la fiabilité des routes descendantes créées par RPL reste un problème ouvert, même en l'absence de nœuds mobiles [16]. J'aimerais m'attaquer à ce problème par le biais du *Segment Routing* [87] qui permet un contrôle complet des chemins de routage. En outre, il permet de définir des traitements spécifiques par messages sur des nœuds cibles. Cela permet de déployer des mécanismes temporaires de reroutage rapide afin de laisser le protocole de routage converger. En combinaison avec un protocole MAC orienté récepteur, l'idée générale est de proposer une gestion de la mobilité orientée réseau dans laquelle l'infrastructure fixe réalise les opérations nécessaires pour maintenir les communications depuis et vers des nœuds mobiles.

## D.4 Dépasser les limites des protocoles asynchrones

Les travaux présentés dans cette section sont issus des travaux de thèse de M. Sébastian L. Sampayo (2018 - 2021) co-encadré avec le professeur Thomas Noël.

### D.4.1 Contexte de recherche

Nos travaux précédents nous ont amenés aux limites des protocoles asynchrones et des gains supplémentaires passeront potentiellement par de nouveaux paradigmes de communication. Pour réduire l'empreinte énergétique des nœuds, la radio est soumise à un cycle de travail qui alterne les périodes de veille et d'activité ce qui nécessite de trouver un compromis entre les délais et la consommation d'énergie. L'émergence des *Wake-Up radios* promet la fin d'un tel compromis.

Une Wake-Up radio se traduit par l'ajout d'un second récepteur radio à très basse consommation (quelques $\mu W$ contre quelques dizaines de $mW$ pour une radio traditionnelle). Une consommation si faible permet de conserver ce récepteur en activité permanente et ainsi d'établir des communications purement asynchrones. Néanmoins, cette technologie présente des contraintes assez fortes qui s'expriment principalement par une portée radio faible, une sensibilité accrue aux interférences, et un débit faible. Ces limitations réservent l'usage de cette technologie au plan de contrôle en complément d'une radio traditionnelle dont l'usage sert le plan de données. La figure D.13 illustre cet usage complémentaire. Lorsqu'un nœud souhaite émettre une donnée, il transmet en premier lieu un message de contrôle (nommé *Wake-Up Signal - WUS*) sur le canal Wake-Up. Lors de sa réception, la destination réveille sa radio principale afin d'être en mesure de recevoir les données transmises via le canal principal. En fin de transmission, la radio principale de la source et de la destination retourne en veille.

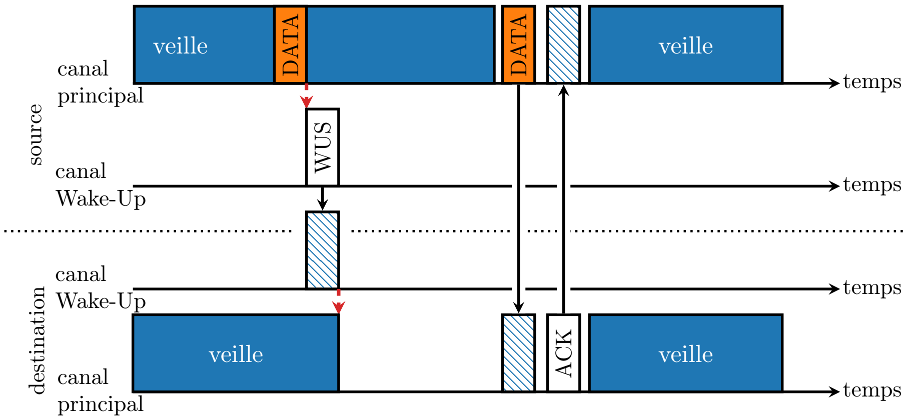



**Figure D.13:** Exemple d'une transmission purement asynchrone à l'aide de Wake-Up radios

Dans ce contexte, le principal défi scientifique consiste à tenir compte de la différence de portée entre les Wake-Up radios et la radio principale. Plus formellement, on peut considérer le graphe $G_1 = (V_1, E_1)$ formé par les liens Wake-Up et le graphe $G_2 = (V_2, E_2)$ formé par les liens de la radio principale où $V_1 = V_2$ et $E1 \subset E_2$. Le graphe $G_1$ étant connexe, nous pouvons également affirmer que $\forall e = (x, y) \in E_2 \mid \exists$ un chemin $p(x, y)$ sur $G_1$ entre les sommets $x$ et $y$. Il faut donc être en mesure de relayer les messages WUS pour minimiser le nombre de transmissions sur la radio principale et donc maximiser les performances. En outre, nous avons démontré que les performances sont significativement améliorées jusqu'à l'utilisation successive de 3 relais avant d'observer une dégradation au-delà [93]. À travers des expérimentations ad hoc, nous avons également mis en lumière la corrélation entre la fiabilité des liens Wake-Up et les performances globales [131] : la modulation simpliste rend les communications Wake-Up très sensibles aux interférences et le débit faible augmente les temps de transmission, renforçant la probabilité d'erreurs. Nous avons proposé un nouveau protocole MAC et des mécanismes inter-couches pour démontrer l'intérêt des Wake-Up radios et défini un nouveau protocole de routage réactif pour relever le défi posé par les portées asymétriques. Ces contributions sont présentées dans la section suivante.

### D.4.2 Contributions

#### W2M : un protocole MAC multi-canal basé sur les Wake-Up radios

Nous avons défini le protocole W2M qui tire parti des Wake-Up radios. Le principe est très proche du protocole présenté sur la figure D.13 à la différence que la destination initie la transmission des données via la diffusion d'un message de contrôle sur la radio principale. Cela évite de déclencher l'émission des données sur expiration d'un temporisateur, dont le délai sera directement proportionnel au nombre de relais Wake-Up traversés. Pour augmenter la capacité du réseau, les messages WUS contiennent le canal radio qui sera utilisé pour la transmission effective des données sur le canal principal. Les messages WUS contiennent donc 3 champs : l'identifiant de la destination finale, l'identifiant du prochain relais Wake-Up et le canal radio.

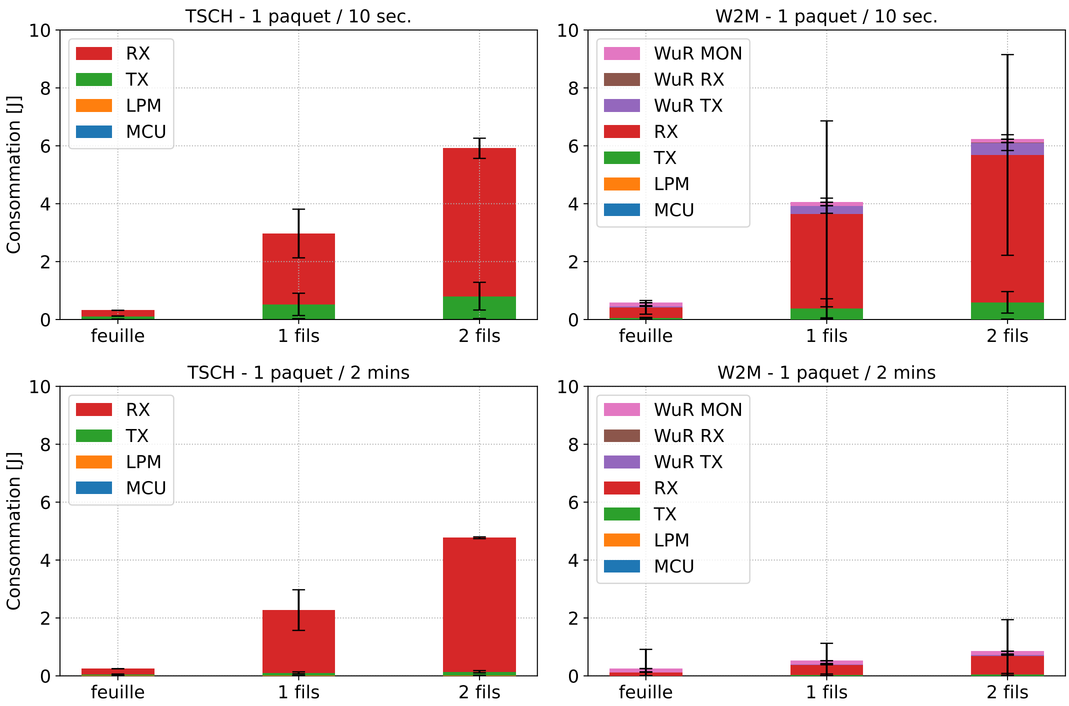



**Figure D.14:** Consommation d'énergie en fonction du type de trafic

Cette contribution a été évaluée par simulation sur Contiki/Cooja et a été comparée avec le protocole synchrone *Time-Slotted Channel Hopping (TSCH)* dont l'ordonnancement est réalisé par Orchestra [95]. L'ensemble des paramètres de simulation est disponible dans [96]. Les résultats montrent que pour une baisse négligeable de la fiabilité des communications (de l'ordre de 2%), W2M réduit et stabilise significativement les délais de bout en bout. Les communications étant purement asynchrones, une source peut envoyer ses données sans temps d'attente. Avec TSCH, si une source rate son opportunité de transmission, elle doit alors attendre la suivante ce qui ajoute 300 ms dans la configuration utilisée. De plus, l'ordonnancement n'étant pas corrélé au trafic, certaines sources ne disposent pas de suffisamment d'opportunités de transmission par fenêtre de temps pour relayer les données issues de leurs sous-graphes, ce qui contribue à augmenter les délais observés. Ce constat est également visible sur la figure D.14 sur laquelle la consommation d'énergie pour TSCH est relativement constante quel que soit le trafic considéré (1 paquet / 10 sec. ou 1 paquet / 2 mins). Cette contre-performance provient de l'ordonnancement qui impose à chaque relais d'ouvrir périodiquement des fenêtres de réception même en l'absence de trafic, ce qui stabilise la consommation d'énergie à un niveau élevé. Avec W2M la consommation d'énergie est directement corrélée au trafic échangé. Nous démontrons ici qu'un ordonnancement statique ne permet pas de tirer les meilleures performances d'un protocole synchrone tel que TSCH, et que dans ce contexte un protocole purement asynchrone tel que W2M permet d'obtenir de meilleurs résultats. La définition d'un ordonnancement optimal et dynamique reste néanmoins un problème ouvert.

### D.4.3 Équilibrage de charge dans RPL

Le protocole de routage RPL présente des défis encore non résolus tels qu'une sélection inefficace des parents, un temps de convergence long suite à la panne d'un parent et une inéquité énergétique entre les nœuds. Pour résoudre ces défis, nous proposons ici de définir des mécanismes inter-couches (entre les couches MAC et réseau) tirant partie des Wake-Up radios pour aider dans la décision finale de routage. En outre, nous avons adapté notre précédente contribution MT-RPL à un contexte Wake-Up radios afin de permettre, en fonction du rang RPL des nœuds, de jouer le rôle de relais opportunistes en lieu et place du parent préféré déterminé par RPL. Dans LoBaPS [98], les relais opportunistes peuvent se déclarer après un délai aléatoire, ce qui permet d'équilibrer la charge mais ne tient pas compte de l'efficacité énergétique. Par conséquent, nous avons proposé eLoBaPS [97] dans lequel le délai de réponse des nœuds opportunistes est proportionnel à l'énergie consommée. Pour éviter que ces délais augmentent indéfiniment (les nœuds continuant à consommer de l'énergie au cours de la vie du réseau), nous maintenons ces délais stables dans le temps via l'équation :

$$B_j(t) = K[e_j(t) - e_{dj}(t)] + C$$

où $B_j(t)$ correspond au délai de réponse du nœud $j$ au temps $t$, $e_j(t)$ est l'énergie consommée par le nœud $j$ au temps $t$ et $e_{dj}(t)$ est l'énergie consommée au temps $t$ par le voisin ayant consommée le moins d'énergie. Enfin, $K$ permet d'ajuster le délai en ms et $C$ est un délai additionnel aléatoire.

Ces contributions ont été évaluées par simulation sur Contiki/Cooja et ont été comparées avec une utilisation simple des Wake-Up radios au niveau MAC (telle que présentée sur la figure D.13) couplée avec RPL. Cette solution est nommée *W-MAC* dans la suite. Pour LoBaPS et eLoBaPS, RPL est utilisé pour créer les routes et rangs initiaux. Après convergence, le routage est entièrement assuré par LoBaPS et eLoBaPS. Pour l'ensemble des solutions, nous avons réduit la portée de la radio principale afin de la faire correspondre à celle des Wake-Up radios. L'ensemble des paramètres de simulation est disponible dans [98] et [97]. La figure D.15 présente l'évolution du taux de livraison. On peut observer qu'il baisse dès la première panne d'un nœud, ce qui arrive dans l'intervalle [750; 1200] sec. pour W-MAC et à 1200 sec. pour nos contributions. Après cet événement, le taux de livraison baisse brutalement et de manière instable alors que LoBaPS conserve un taux de livraison stable jusqu'à la première panne d'un nœud et une baisse plus stable ensuite. eLoBaPS surpasse LoBaPS en réduisant la durée du déclin du taux de livraison jusqu'à la perte de connexité du graphe. La figure D.16 présente le gain offert par LoBaPS et eLoBaps sur la durée de vie du réseau par rapport à W-MAC. La durée de vie est calculée entre le démarrage du réseau jusqu'au moment où un nœud tombe en panne faute d'énergie. En cherchant à équilibrer la consommation énergétique à chaque saut, eLoBaPS permet d'augmenter la durée de vie de 17% par rapport à LoBaPS et de 38% par rapport à W-MAC.

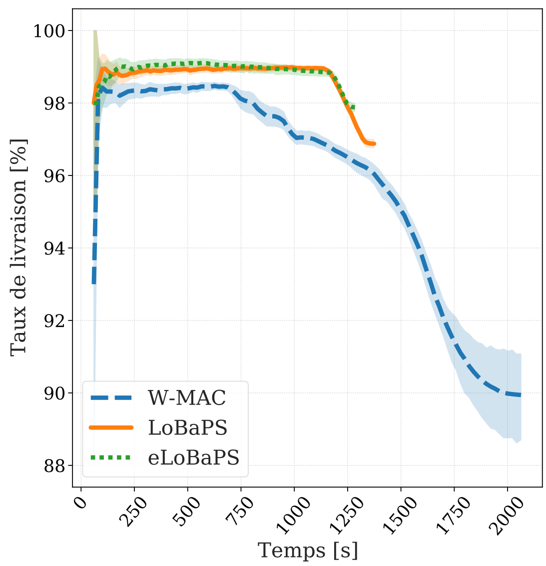



**Figure D.15:** Taux de livraison pour un trafic de 1 paquet / 10 sec.

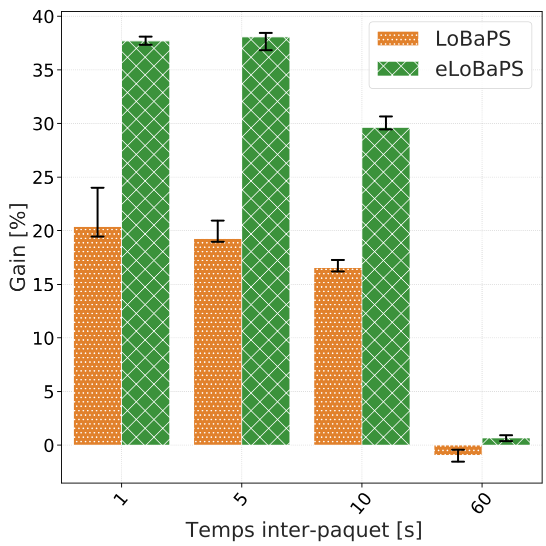



**Figure D.16:** Gain sur la durée de vie par rapport à W-MAC

Nous avons démontré ici que l'utilisation des Wake-Up radios permet d'améliorer les performances des réseaux asynchrones, non seulement au niveau MAC mais également au niveau réseau. Cependant, nous avons fait ici l'hypothèse que la portée entre les deux radios était identique. Il nous reste encore à lever cette hypothèse pour finaliser nos propositions et offrir un ensemble réaliste de protocoles basé sur les Wake-Up radios.

### REFLOOD : un protocole de routage réactif

Nous avons étudié le routage des messages Wake-Up en considérant que le graphe $G_2$ formé par la radio principale est un graphe complet composé de $n$ sommets et de $\frac{n(n-1)}{2}$ arcs et que le graphe $G_1$ formé par la radio Wake-Up est connexe et formé des même $n$ sommets. Dans ce contexte, nous avons proposé le protocole réactif REFLOOD [103]. Lorsque le nœud $n_i$ souhaite transmettre un message au nœud $n_j$ avec $i \neq j$, il inonde le graphe $G_1$ avec un message WUS qui contient l'adresse de $n_j$. Lors de sa réception, le nœud $n_j$ allume sa radio principale ce qui permet aux nœuds $n_i$ et $n_j$ d'échanger les données et l'acquittement correspondant sur $G_2$. L'inondation est contrôlée en limitant sa portée (après un certain nombre de sauts les messages WUS sont éliminés) et en bloquant localement l'inondation pendant un certain délai après la rediffusion du premier WUS. Bien évidemment, ni $n_i$ (la source) ni $n_j$ (la destination) ne participent à l'inondation.

REFLOOD a été évalué par simulation sur Contiki/Cooja et a été comparé avec le protocole MAC ContikiMAC [33] (le graphe $G_2$ étant complet, aucun protocole de routage n'est nécessaire si les communications utilisent exclusivement $G_2$) et PROPL, un protocole de routage pro-actif pour lequel nous avons réduit la portée de la radio principale pour la faire correspondre à celle de la Wake-Up radio. En agissant ainsi, $G_1$ et $G_2$ deviennent un unique et même graphe sur lequel le routage est assuré par RPL. L'ensemble des paramètres de simulation est disponible dans [103]. La figure D.17 présente le délai de bout en bout. On y observe que REFLOOD présente, pour un même débit Wake-Up, des délais inférieurs à ceux obtenus par ContikiMAC et PROPL. Les protocoles de routage réactif sont souvent décriés en raison de délais plus importants dus à la mise en place des routes. Dans REFLOOD, ce délai est compensé par l'utilisation de chemins multiples alors qu'avec PROPL les messages WUS passent du temps en file d'attente en raison de l'utilisation de chemins uniques. L'usage de chemins multiples permet également à REFLOOD de mieux équilibrer la charge, ce qui augmente la durée de vie globale du réseau, comme illustré sur la figure D.18.

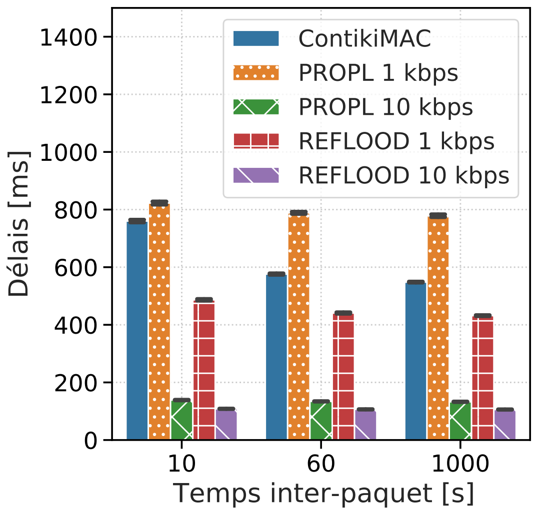



**Figure D.17:** Délais de bout en bout

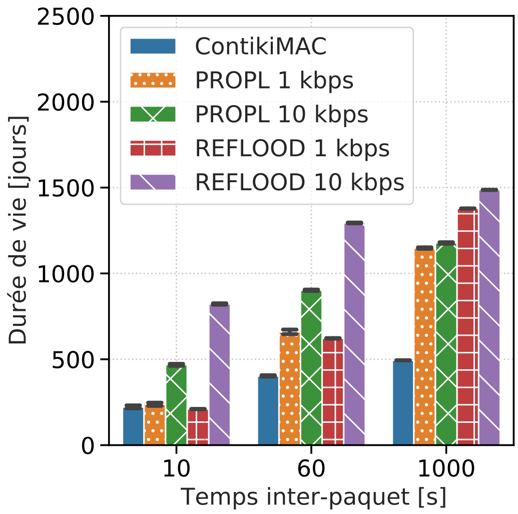



**Figure D.18:** Durée de vie du réseau

REFLOOD constitue une première étape vers un protocole de routage sur une infrastructure double telle que celle formée par la radio principale et la radio Wake-Up. En outre, l'usage d'une stratégie réactive me paraît plus adaptée pour s'accommoder des limitations liées aux Wake-Up radios. Elle permet également de bénéficier de chemins multiples ce qui augmente le taux de livraison, la durée de vie du réseau et réduit les délais. Néanmoins, le travail continue afin d'étendre le routage à des graphes $G_2$ quelconques.

### D.4.4 Conclusion et perspectives

L'usage des Wake-Up radios dans les réseaux LLWN permet de bénéficier de communications purement asynchrones, ce qui permet une réduction significative de la consommation énergétique et des délais, comme nous l'avons montré avec notre protocole W2M. Les gains peuvent également s'étendre au niveau réseau en utilisant le canal Wake-Up comme un canal de contrôle additionnel. Néanmoins, le défi principal de cette technologie repose sur la faible portée radio offerte, surtout en comparaison de la radio principale. Dans ce contexte, REFLOOD constitue une première proposition qui tient compte de cette asymétrie pour calculer des routes sur infrastructure à deux niveaux.

La perspective la plus encourageante repose sur l'extension de nos derniers travaux à des graphes $G_2$ (donc formés par la radio principale) quelconques. Dans un premier temps, nous pouvons étendre RELOOD en un protocole hybride. Avec une stratégie pro-active, le graphe $G_2$ peut être décomposé en des cliques hiérarchiques interconnectées via des routeurs de bordures (donc appartenant à au moins deux cliques). Ensuite, une stratégie réactive permet de réveiller l'un des routeurs de bordure au sein d'une même clique. Les données peuvent ainsi être acheminées de cliques en cliques jusqu'à atteindre la destination finale. Une alternative consiste à fusionner $G_1$ et $G_2$ en un graphe unique multivalué dans lequel les liens sont ceux de $G_2$ élagués des liens $(x_i, x_j)$ avec $i \neq j$ pour lesquels la longueur du chemin dans $G_1$ entre $x_i$ et $x_j$ est supérieure à un seuil. Nous pouvons ensuite calculer les chemins qui minimisent le coût issu de $G_2$ et utiliser le coût du chemin issu de $G_1$ pour départager les cas d'égalité.

## D.5 Conclusion et projet de recherche

L'intégration d'entités mobiles au sein des réseaux sans fil contraints (LLWN) permet d'en étendre les domaines d'application, allant des transports intelligents à la santé. En outre, le nombre d'objets connectés devrait atteindre 125 milliards d'ici 2030 [2]. Dans un tel contexte, les réseaux LLWN constitueront l'un des éléments clefs de l'Internet du futur. Néanmoins, sans support spécifique, la mobilité dégrade la qualité des communications et engendre des pertes, des déconnexions, et une augmentation des délais. Nous avons identifié trois principaux défis : déterminer la destination d'une source mobile, minimiser la durée des communications issues d'une source mobile, et déterminer la position logique d'une destination mobile. De mon point de vue, nous avons résolu les deux premiers défis à travers notre contribution X-Machiavel. Cette dernière permet aux nœuds mobiles de bénéficier de relais opportunistes ou de voler le canal pré-réservé par un nœud fixe. Ces travaux ont également mis en lumière les limites des protocoles asynchrones que nous avons cherché à repousser en explorant la technologie Wake-Up radio. Nous avons montré, à travers notre contribution W2M, que l'utilisation de cette technologie permet de définir des protocoles MAC purement asynchrones dont les performances sont équivalentes (voire supérieures dans certains scénarios) à celles proposées par des protocoles synchrones. Néanmoins, les Wake-Up radios présentent des contraintes fortes dont il faut s'accommoder telles que la très faible portée radio en comparaison d'une radio classique. Nous avons étudié ce problème et proposé le protocole de routage réactif REFLOOD en faisant l'hypothèse que le graphe formé par les liens établis par la radio principale est un graphe complet. Par la suite nous avons évoqué des pistes prometteuses pour étendre ce travail à des graphes quelconques. Pour résoudre le troisième et dernier défi, nous avons étudiés deux pistes opposées. Masquer la position réelle des nœuds mobiles à travers l'utilisation d'un agent relais et de tunnels constitue notre première piste, mais elle introduit un point de défaillance unique dans le réseau et surtout un routage sous-optimal. Pour ces raisons nous avons privilégié le fait d'expliciter la position courante des nœuds mobiles avec des mises à jour du plan de données. Notre solution inter-couches MT-RPL favorise la dynamique des nœuds mobiles mais à une incidence sur les routes descendantes qui deviennent instables. Il est encore nécessaire de définir des mécanismes de reroutage rapide pour consolider cette contribution.

Depuis les années 2000, de nombreux protocoles de communication et de technologies ont été proposés. Il est aujourd'hui difficile de choisir un ensemble cohérent de protocoles qui réponde aux besoins d'une ou plusieurs applications parmi cette offre pléthorique. Dans un second temps, la configuration de cet ensemble de protocoles reste également complexe, et il est souvent nécessaire de procéder de manière empirique pour trouver la configuration idéale pour une application donnée. Afin d'apporter plus de flexibilité dans les réseaux de communication, je souhaite passer d'une pile protocolaire monolithique héritée du modèle TCP/IP à des fonctions réseaux programmables. L'émergence des réseaux programmables, l'adoption de l'informatique en nuage et les avancées dans l'intelligence artificielle représentent des opportunités pour changer définitivement comment des objets connectés interagissent. Dans un premier temps, nous pouvons tirer profit des réseaux programmables pour optimiser de manière continue les réseaux LLWN et ainsi rendre les réseaux autonomes. En parallèle, je propose de construire une pile protocolaire entièrement programmable en réutilisant les concepts de la programmation logicielle : les nœuds qui composent le réseau sont assimilés à des « processeurs réseaux » munis d'un ensemble d'instructions (envoyer une trame, armer un temporisateur, etc.) et exécutent des programmes (la chaîne de traitement des données). À terme, ces derniers seront générés automatiquement puis évalués sur des reproductions virtuelles avant déploiement réel. Ce programme ambitieux de recherche sera au cœur de nos activités de recherche dans les prochaines années.